

# Adaptive Regelgenerierung und ihre Verwendung zur Diagnose des septischen Schocks

Dissertation  
zur Erlangung des Doktorgrades  
der Naturwissenschaften

vorgelegt beim Fachbereich Biologie und Informatik  
der Johann Wolfgang Goethe-Universität  
in Frankfurt am Main

von  
Dipl.-Math. Jürgen Paetz  
aus Offenbach am Main

Frankfurt am Main 2002  
(DF1)

vom Fachbereich Biologie und Informatik der  
Johann Wolfgang Goethe-Universität als Dissertation angenommen.

Dekan: Prof. Dr. Bruno Streit

Gutachter: PD Dr. habil. Rüdiger Brause, Prof. Dott. Ing. Roberto Zicari

Datum der Disputation: 2. September 2002

# Vorwort

Die Möglichkeit zur Abfassung eines Vorwortes möchte ich nutzen, um mich bei Herrn PD Dr. habil. Rüdiger Brause, Herrn Prof. Dr. Dr. med. Ernst Hanisch und der Deutschen Forschungsgemeinschaft zu bedanken, bei ersteren für die Initiierung des DFG-Projekts MEDAN („Medizinische Datenanalyse mit Neuronalen Netzen - Analyse des septischen Schocks: Einfluss auf Therapie und Prognose“, HA1456/7-1, [www.medan.de](http://www.medan.de)), bei letzterer für die Bereitstellung der Mittel. Bezüglich der Abfassung dieser Arbeit möchte ich mich insbesondere bei Herrn PD Dr. habil. Rüdiger Brause für seine dauerhafte Diskussionsbereitschaft und Unterstützung bedanken. Bedanken möchte ich mich auch bei den beiden MEDAN-Projektmitarbeitern Herrn Dipl.-Inform. Björn Arlt für die Erweiterung und Betreuung der MEDAN-Datenbank und Herrn Dr.-Ing. Fred Hamker für allerlei anregende Diskussionen innerhalb des MEDAN-Projekts sowie bei den Projektbeteiligten seitens der Medizin für die gute Zusammenarbeit und die wertvollen medizinischen Hinweise im Rahmen des MEDAN-Projekts. Ein Dankeschön geht auch an alle Kliniken, die dem Projekt ihre Daten zur Verfügung gestellt haben und an alle Dokumentare, die Daten aus den Patientenakten in die MEDAN-Datenbank übertragen haben. Gedankt sei auch der Hermann-Willkomm-Stiftung, die die aktive Teilnahme an einer Konferenz in San Jose, CA, USA, finanziell unterstützt hat.

Ein besonderes Dankeschön geht an meinen privaten Freundeskreis.

Das Anliegen dieser Dissertation<sup>1</sup> ist eine intensive Beschäftigung mit adaptiven Datenanalyseverfahren und eine Wertschöpfung für die medizinische Anwendung unter Berücksichtigung der Grundlagen sowie eine Werbung für die interdisziplinäre Arbeit im Rahmen der Angewandten Informatik, in meinem Fall eine Zusammenarbeit mit dem Zentrum für Chirurgie am Fachbereich Medizin der J.W. Goethe-Universität Frankfurt am Main.

Frankfurt am Main, im Februar 2002

---

<sup>1</sup>Angefertigt wurde die Arbeit unter WinEdt 1.414 und MikTeX 1.10b; Grafiken wurden mit Microsoft Word 97, Corel Draw 7 und Matlab 5.3 erzeugt. Die Experimente wurden auf einem handelsüblichen PC (Intel Pentium II 333MHz/256MB RAM/zwei 4GB Festplatten/Windows NT 4.0) durchgeführt.

„Automat und Telespiel  
leiten heute die Zukunft ein  
Computer für den Kleinbetrieb  
Computer für das Eigenheim  
Reisen – Zeit – Medizin – Unterhaltung  
Reisen – Zeit – Medizin – Unterhaltung  
Computerwelt – Computerwelt  
denn Zeit ist Geld“

Kraftwerk, *Computerwelt*, 1981

# Inhaltsverzeichnis

<b>Vorwort</b>	<b>iii</b>
<b>Inhaltsverzeichnis</b>	<b>v</b>
<b>Abbildungsverzeichnis</b>	<b>ix</b>
<b>Tabellenverzeichnis</b>	<b>xiii</b>
<b>Symbolverzeichnis</b>	<b>xix</b>
<b>Abkürzungsverzeichnis</b>	<b>xxiii</b>
<b>1 Einleitung</b>	<b>1</b>
<b>2 Adaptive Datenanalyse in der Medizin</b>	<b>7</b>
2.1 Datenanalyse in der Medizin . . . . .	7
2.1.1 Medizinische Datenbasen . . . . .	9
2.1.2 Elementare medizinische Standardanalysen . . . . .	10
2.1.3 Score-Systeme . . . . .	11
2.1.4 Studie - Multizentrenstudie . . . . .	12
2.1.5 Alarmsysteme . . . . .	14
2.2 Neuronale Netze . . . . .	15
2.2.1 Grundlagen - Aufbau neuronaler Netze . . . . .	15
2.2.2 Backpropagation . . . . .	17
2.2.3 Radiale-Basis-Funktionen-Netze . . . . .	18
2.2.4 Selbstorganisierende Karten und U-Matrix-Methode . . . . .	22
2.2.5 (Wachsende) Neuronale Gase . . . . .	26
2.3 Zustandsanalysen . . . . .	27
2.3.1 Clusteranalyse . . . . .	27
2.3.2 Klassifikation . . . . .	29
2.3.3 Zeitreihenanalyse . . . . .	33
2.4 Abhängigkeitsanalysen . . . . .	35
2.4.1 Signale . . . . .	36
2.4.2 Statistische Standardverfahren . . . . .	37
2.4.3 Lineare ICA . . . . .	40

2.4.4	Modellierung . . . . .	44
2.5	Regelgenerierung . . . . .	46
2.5.1	Neuro-Fuzzy-Regelgenerierung . . . . .	47
2.5.2	Regelgenerierung im Maschinellen Lernen . . . . .	54
2.5.3	Kombination von Regelmengen . . . . .	61
<b>3</b>	<b>Regelgenerierung mit Fuzzy-RecBF-DDA-Netzen</b>	<b>65</b>
3.1	Das Fuzzy-RecBF-DDA-Netz und Verbesserungen . . . . .	66
3.2	Experimente auf Benchmark-Daten . . . . .	76
3.2.1	Experiment mit den IRIS-Daten . . . . .	77
3.2.2	Experiment mit den Cancer1-Daten . . . . .	79
3.2.3	Experiment mit den Diabetes1-Daten . . . . .	82
3.2.4	Experiment mit den Spiraldaten . . . . .	84
3.2.5	Bewertung der Ergebnisse . . . . .	87
3.3	Behandlung fehlender Werte . . . . .	90
3.4	Gewinnung scharfer Regeln . . . . .	93
3.5	Die Maße „Wichtigkeit“ und „Frühzeitigkeit“ . . . . .	103
3.6	Regelgewinnung mit dem Fuzzy-WvRecBF-DDA-Netz . . . . .	108
3.7	Ähnlichkeit von Fuzzy-RecBF-DDA-Regeln . . . . .	112
<b>4</b>	<b>Erzeugen von Generalisierungsregeln</b>	<b>117</b>
4.1	Generalisierungsregeln durch Durchschnittsbildung . . . . .	118
4.2	Eigenschaften der durchschnittsbasierten Regeln . . . . .	121
4.3	Klassifikation mit einer symbolischen Regelmenge . . . . .	132
4.4	Ergebnisse auf Benchmark-Daten . . . . .	133
4.5	Die Maße „Wichtigkeit“ und „Frühzeitigkeit“ . . . . .	135
4.6	Auswahlkriterien . . . . .	138
4.7	Der innere und äußere Generalisierungsindex . . . . .	140
4.8	Vergleich zu Assoziationsregeln . . . . .	142
4.9	Ähnlichkeit von Generalisierungsregeln . . . . .	144
<b>5</b>	<b>Die metrischen Patientendaten der Vorstudie</b>	<b>147</b>
5.1	SIRS - Sepsis - septischer Schock . . . . .	147
5.2	Vorverarbeitung der Daten . . . . .	150
5.3	Deskriptive Statistik . . . . .	153
5.4	Zustandsanalyse . . . . .	158
5.5	Regelgenerierung . . . . .	161
5.5.1	Regelgenerierung mit Fuzzy-RecBF-DDA-Netzen . . . . .	162
5.5.2	Manuelle und automatische Regelgenerierung . . . . .	171
<b>6</b>	<b>Ausgewählte symbolische Daten</b>	<b>175</b>
6.1	Beschreibung der verwendeten Daten . . . . .	175
6.2	Anwendung des Generalisierungsalgorithmus . . . . .	176
6.3	Generalisierung und Wichtigkeit . . . . .	181

6.4	Generalisierung und Ähnlichkeit . . . . .	185
6.5	Generalisierung und Generalisierungsindices . . . . .	188
6.6	Vergleich mit Assoziationsregeln und FP-Bäumen . . . . .	190
<b>7</b>	<b>Die septischer Schock-Klinikdaten</b>	<b>195</b>
7.1	Vorverarbeitung und deskriptive Statistik . . . . .	195
7.2	Regelgenerierung mit Fuzzy-RecBF-DDA-Netzen . . . . .	200
7.2.1	Datensätze und Hauptergebnisse . . . . .	202
7.2.2	Detaillierte Klassifikations- und Regelergebnisse . . . . .	214
7.3	Erzeugung von Generalisierungsregeln . . . . .	231
7.3.1	Generalisierung mit allen 145 Items . . . . .	231
7.3.2	Generalisierung mit den wichtigsten 35 Items . . . . .	235
7.4	Kombination von Regelmengen . . . . .	242
7.5	Aufbau eines Alarmsystems . . . . .	247
7.5.1	Erzeugung der Alarme . . . . .	248
7.5.2	Alarmhistorien auf den Klinikdatensätzen . . . . .	249
7.5.3	Zusammenfassende Diskussion . . . . .	257
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>259</b>
<b>A</b>	<b>Daten und Analysesoftware</b>	<b>263</b>
A.1	Benchmark-Daten . . . . .	263
A.2	Vorstudiendaten . . . . .	263
A.3	Klinikdaten . . . . .	268
A.4	Ausgewählte symbolische Klinikdaten . . . . .	269
A.5	Analysesoftware . . . . .	269
<b>B</b>	<b>ROC-Kurven</b>	<b>273</b>
<b>C</b>	<b>Tabellen</b>	<b>279</b>
<b>D</b>	<b>Das Fuzzy-RecBF-DDA-Netz von Huber/Berthold</b>	<b>291</b>
	<b>Literaturverzeichnis</b>	<b>295</b>
	<b>Stichwortverzeichnis</b>	<b>313</b>
	<b>Lebenslauf</b>	<b>317</b>



# Abbildungsverzeichnis

2.1	Datenanalyse-Prozess in der Medizin. . . . .	8
2.2	Abstrakte Zustände innerhalb einer Patientenkurve. . . . .	14
2.3	Architektur eines (überwachten) Feedforward-Netzes. . . . .	16
2.4	Architektur des RBF-DDA-Netzes. . . . .	19
2.5	Das Schema einer SOM. . . . .	23
2.6	U-Matrix zweier getrennt liegender 3-dim. Normalverteilungen. . . . .	25
2.7	Schema einer Clusterung mit einer SOM. . . . .	29
2.8	Schema einer Klassifizierung im zweidimensionalen Raum. . . . .	30
2.9	Schema einer Zeitreihenvorhersage mit einem neuronalen Netz. . . . .	34
2.10	Schema einer nichtlinearen Regression. . . . .	37
2.11	Schema einer linearen und nichtlinearen PCA. . . . .	39
2.12	Drei ungemischte Signale. . . . .	41
2.13	Drei linear gemischte Signale. . . . .	41
2.14	Drei Signale nach einer PCA. . . . .	41
2.15	Drei Signale nach einer LICA. . . . .	42
2.16	Problem der Regelgenerierung aus einer Ellipse. . . . .	48
2.17	Regelerzeugung durch Clusterung und durch Klassifikation. . . . .	48
2.18	Eine 2-dim. trapezoide Zugehörigkeitsfunktion. . . . .	50
2.19	Prinzip der Assoziationsregeln und Generalisierungsregeln. . . . .	55
3.1	Fuzzy-RecBF-DDA (Beispiel): Einfügen des ersten Punktes. . . . .	69
3.2	Fuzzy-RecBF-DDA (Beispiel): Einfügen des zweiten Punktes. . . . .	70
3.3	Fuzzy-RecBF-DDA (Beispiel): Einfügen des dritten Punktes. . . . .	70
3.4	Fuzzy-RecBF-DDA (Beispiel): Einfügen des vierten Punktes. . . . .	71
3.5	Fuzzy-RecBF-DDA (Original): Unerwünschte Überlappung von Core- Bereichen zweier verschiedener Klassen im Originalalgorithmus. . . . .	72
3.6	Generierte Regeln aus den Spiraldaten (1. Versuch), <i>nearby</i> = 2. . . . .	84
3.7	Generierte Regeln aus den Spiraldaten (2. Versuch), <i>nearby</i> = 2. . . . .	85
3.8	Generierte Regeln aus den Spiraldaten (1. Versuch), <i>nearby</i> = 0.14. . . . .	85
3.9	Generierte Regeln aus den Spiraldaten (2. Versuch), <i>nearby</i> = 0.14. . . . .	86
3.10	Fuzzy-RecBF-DDA (Beispiel): Ersetzung eines fehlenden Wertes. . . . .	91
3.11	Konstruktion scharfer Regeln. . . . .	94
3.12	IRIS-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängig- keit vom Fuzzy-Niveau. . . . .	96

3.13	IRIS-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	96
3.14	Cancer1-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	97
3.15	Cancer1-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	97
3.16	Diabetes1-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	98
3.17	Diabetes1-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	98
3.18	Spiral-Daten ( $nearby = 0.14$ ): Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	99
3.19	Spiral-Daten ( $nearby = 0.14$ ): Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	100
3.20	Spiral-Daten ( $nearby = 2$ ): Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	100
3.21	Spiral-Daten ( $nearby = 2$ ): Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	101
3.22	Diabetes1-Daten: Verteilung der optimalen scharfen Regeln der Klasse 1. . . . .	102
3.23	Diabetes1-Daten: Verteilung der optimalen scharfen Regeln der Klasse 2. . . . .	102
4.1	Beweis der Leveleigenschaft. . . . .	122
4.2	Beweis der Optimalität der Gütemaße. . . . .	125
4.3	Klassifikationsergebnisse auf dem IRIS-Datensatz bei heuristischem Pruning. . . . .	135
5.1	Anzahl der Patienten aus der Studie [WBH98] und der gesamten Vorstudie. . . . .	149
5.2	Häufig12Vor-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	164
5.3	Häufig12Vor-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	165
5.4	Wichtig17Vor-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	169
5.5	Wichtig17Vor-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	169
6.1	ASK-Daten: Gewichtete Konfidenz in Abhängigkeit von der Frühzeitigkeit für die Klasse „überlebt“. . . . .	185
6.2	ASK-Daten: Gewichtete Konfidenz in Abhängigkeit von der Frühzeitigkeit für die Klasse „verstorben“. . . . .	186

6.3	ASK-Daten: Gewichtete Konfidenz in Abhängigkeit von der Frühzeitigkeit für beide Klassen. . . . .	186
6.4	ASK-Daten: Dendrogramm für die Unähnlichkeit der erzeugten Generalisierungsregeln. . . . .	187
7.1	Klinikdaten: Füllgrad der Daten pro Patient. . . . .	197
7.2	Häufig16K-Daten: ROC-Flächeninhalt in der Abhängigkeit vom Zeitraum. . . . .	209
7.3	Klinikdaten: ROC-Flächeninhalte. . . . .	209
7.4	Klinikdaten: ROC-Flächeninhalte der Subsystem-Datensätze. . . . .	210
7.5	Klinikdaten: ROC-Flächeninhalte der Score-Datensätze. . . . .	211
7.6	SystDiasThrombo3K: Visualisierung des Datensatzes. . . . .	212
7.7	Häufig16K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	216
7.8	Häufig16K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	216
7.9	SOFA1K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	220
7.10	SOFA1K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	220
7.11	SystThrombo2K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	222
7.12	SystThrombo2K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	223
7.13	Wichtig6K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	225
7.14	Wichtig6K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	226
7.15	SystDiasThrombo3K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	228
7.16	SystDiasThrombo3K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau. . . . .	229
7.17	SystDiasThrombo3K-Daten: Verteilung der optimalen scharfen Regeln der Klasse ü. . . . .	229
7.18	SystDiasThrombo3K-Daten: Verteilung der optimalen scharfen Regeln der Klasse v. . . . .	230
7.19	Klinikdaten: Gewichtete Konfidenz in Abhängigkeit von der Frühzeitigkeit. . . . .	239
7.20	Klinikdaten: Dendrogramm für die Unähnlichkeit der erzeugten Generalisierungsregeln. . . . .	240
7.21	Schema des kombinierten Systems. . . . .	243
7.22	Häufig16K-Daten: Alarmkennzahlen. . . . .	250
7.23	SOFA1K-Daten: Alarmkennzahlen. . . . .	253
7.24	SystThrombo2K-Daten: Alarmkennzahlen. . . . .	254

7.25	Wichtig6K-Daten: Alarmkennzahlen. . . . .	255
7.26	SystDiasThrombo3K-Daten: Alarmkennzahlen. . . . .	256
A.1	Spiraldaten. . . . .	264
A.2	Normalverteilte Daten mit Ausreißern. . . . .	265
B.1	Cancer1-Daten: ROC-Kurven. . . . .	274
B.2	Diabetes1-Daten: ROC-Kurven. . . . .	274
B.3	Häufig12Vor-Daten und Wichtig17Vor-Daten: ROC-Kurven. . . . .	275
B.4	Häufig16K-Daten: ROC-Kurven. . . . .	275
B.5	SOFA1K-Daten: ROC-Kurven. . . . .	276
B.6	SystThrombo2K-Daten: ROC-Kurven. . . . .	276
B.7	Wichtig6K-Daten: ROC-Kurven. . . . .	277
B.8	SystDiasThrombo3K-Daten: ROC-Kurven. . . . .	277
B.9	SystDiasThrombo3K/Sym35-Daten: ROC-Kurven. . . . .	278
D.1	Eine nicht dokumentierte Situation im Originalalgorithmus. . . . .	293

# Tabellenverzeichnis

3.1	Anzahl der Kombinationen bei einer Regelgenerierung mit Rechtecken.	75
3.2	IRIS-Daten: Korrekte Klassifikationen.	77
3.3	IRIS-Daten: Fehler auf den Trainings- und Testdaten.	77
3.4	IRIS-Daten: Anzahl der erzeugten Regeln pro Klasse.	78
3.5	Cancer1-Daten: Korrekte Klassifikationen.	80
3.6	Cancer1-Daten: Fehler auf den Trainings- und Testdaten.	80
3.7	Cancer1-Daten: Anzahl der erzeugten Regeln pro Klasse.	80
3.8	Diabetes1-Daten: Korrekte Klassifikationen.	83
3.9	Diabetes1-Daten: Fehler auf den Trainings- und Testdaten.	83
3.10	Diabetes1-Daten: Anzahl der erzeugten Regeln pro Klasse.	83
3.11	Spiraldaten ( $nearby = 2$ ): Korrekte Klassifikationen.	86
3.12	Spiraldaten ( $nearby = 2$ ): Fehler auf den Trainings- und Testdaten.	87
3.13	Spiraldaten ( $nearby = 2$ ): Anzahl der erzeugten Regeln pro Klasse.	87
3.14	Nearby-Parameter verschiedener Datensätze.	88
3.15	Spiraldaten ( $nearby = 0.14$ ): Korrekte Klassifikationen.	88
3.16	Spiraldaten ( $nearby = 0.14$ ): Fehler auf den Trainings- und Testdaten.	89
3.17	Spiraldaten ( $nearby = 0.14$ ): Anzahl der erzeugten Regeln pro Klasse.	89
3.18	Diabetes1-Daten (fehlende Werte): Korrekte Klassifikationen.	91
3.19	Diabetes1-Daten (fehlende Werte): Fehler auf den Trainings- und Testdaten.	92
3.20	Diabetes1-Daten (fehlende Werte): Anzahl der erzeugten Regeln pro Klasse.	92
3.21	Cancer1-Daten: Wichtigkeit der Variablen.	104
3.22	Cancer1-Daten: Korrekte Klassifikationen (nur Variablen 2 und 6).	105
3.23	Cancer1-Daten: Korrekte Klassifikationen (nur Variablen 7 und 9).	105
3.24	IRIS-Daten: Wichtigkeit der Variablen.	106
3.25	NormAusreißer-Daten: Korrekte Klassifikationen, Fuzzy-RecBF-DDA.	111
3.26	NormAusreißer-Daten: Anzahl der erzeugten Regeln pro Klasse, Fuzzy-RecBF-DDA.	111
3.27	NormAusreißer-Daten: Korrekte Klassifikationen, Fuzzy-WvRecBF-DDA.	111
3.28	NormAusreißer-Daten: Anzahl der erzeugten Regeln pro Klasse, Fuzzy-WvRecBF-DDA.	112
3.29	Ähnlichkeit der metrischen Regelmengen.	115

4.1	Vote-Daten: Wichtigkeit der Items. . . . .	136
4.2	Breast-Daten: Wichtigkeit der Items. . . . .	137
4.3	Vote-Daten: Neu erzeugte, möglich zu erzeugende Itemsets und äußere Generalisierungsindices. . . . .	141
4.4	ASK-Daten: Neu erzeugte, möglich zu erzeugende Itemsets und ä- ußere Generalisierungsindices. . . . .	141
5.1	Vorstudiendaten: Schema der Daten nach einem 24h-Sampling. . . . .	151
5.2	Vorstudiendaten: Messhäufigkeit nach einem 24h-Sampling für die septischen Schock-Patienten. . . . .	152
5.3	Vorstudiendaten: Zeitreihendynamik der Variablen: vom Arzt erwünsch- te bzw. erwartete Messhäufigkeit und tatsächliche Messhäufigkeit. . . . .	153
5.4	Vorstudiendaten: Korrelationen zwischen zwei Messvariablen mit den höchsten Korrelationsdifferenzen $> 0.5$ zwischen Verstorbenen und Überlebenden (alle Tage, alle Patienten). . . . .	154
5.5	Vorstudiendaten: Anzahl der Datensamples und Signifikanzniveau der Korrelationen der Tab. 5.4. . . . .	155
5.6	Vorstudiendaten: Korrelationen zwischen zwei Messvariablen mit Kor- relationsdifferenzen $> 0.3$ zwischen Verstorbenen und Überlebenden (alle Tage, alle Patienten). . . . .	156
5.7	Vorstudiendaten: Messhäufigkeit der Variablen aus der Tab. 5.6 (alle Patienten der Vorstudiendaten, alle Tage). . . . .	156
5.8	Vorstudiendaten: Anzahl der Datensamples und Signifikanzniveau der Korrelationen der Tab. 5.6. . . . .	157
5.9	Wichtig17Vor-Daten: Korrekte Klassifikationen auf den Testdaten mit SGNG und RBF-DDA. . . . .	159
5.10	Häufig12Vor-Daten: Korrekte Klassifikationen. . . . .	162
5.11	Häufig12Vor-Daten: Fehler auf den Trainings- und Testdaten. . . . .	163
5.12	Häufig12Vor-Daten: Anzahl der erzeugten Regeln pro Klasse. . . . .	163
5.13	Häufig12Vor-Daten (Fuzzy-WvRecBF-DDA): Korrekte Klassifikatio- nen. . . . .	165
5.14	Häufig12Vor-Daten (Fuzzy-WvRecBF-DDA): Anzahl der erzeugten Regeln pro Klasse. . . . .	165
5.15	Häufig12Vor-Daten: Wichtigkeit der Variablen. . . . .	166
5.16	Wichtig17Vor-Daten: Korrekte Klassifikationen. . . . .	167
5.17	Wichtig17Vor-Daten: Fehler auf den Trainings- und Testdaten. . . . .	168
5.18	Wichtig17Vor-Daten: Anzahl der erzeugten Regeln pro Klasse. . . . .	168
5.19	Wichtig17Vor-Daten: Wichtigkeit der Variablen. . . . .	171
6.1	ASK-Daten: Ergebnis-Kennzahlen der erzeugten Generalisierungsre- geln. . . . .	177
6.2	ASK-Daten: Wichtigkeit der Items I. . . . .	179
6.3	ASK-Daten: Wichtigkeit der Items II. . . . .	180

6.4	ASK-Daten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln unter Berücksichtigung der Wichtigkeit. . . . .	182
6.5	ASK-Daten: Wichtigkeit der „wichtigsten“ Items. . . . .	184
6.6	ASK-Daten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln unter Berücksichtigung der Generalisierungsindices. . . . .	189
6.7	ASK-Daten: Mittlere Flächeninhalte der ROC-Kurven. . . . .	190
6.8	ASK-Daten: Ergebnisse des A-priori-Verfahrens. . . . .	192
7.1	Klinikdaten: Messhäufigkeit nach einem 24h-Sampling für die septischen Schock-Patienten und Anzahl der Patienten. . . . .	197
7.2	Klinikdaten: Korrelationen zwischen zwei Messvariablen mit Korrelationsdifferenzen $> 0.5$ zwischen Verstorbenen und Überlebenden (alle Tage, alle Patienten). . . . .	201
7.3	Klinikdaten: Messhäufigkeit der Variablen aus der Tab. 7.2 (alle Patienten der Klinikdaten, alle Tage). . . . .	201
7.4	Klinikdaten: Anzahl der Datensamples und Signifikanzniveau der Korrelationen der Tab. 7.2. . . . .	201
7.5	Klinikdaten: Anzahl der verwendeten Samples. . . . .	204
7.6	Klinikdaten: Klassifikationsleistung auf verschiedenen Datensätzen. . . . .	208
7.7	Häufig16K-Daten: Korrekte Klassifikationen. . . . .	214
7.8	Häufig16K-Daten: Fehler auf den Trainings- und Testdaten. . . . .	214
7.9	Häufig16K-Daten: Anzahl der erzeugten Regeln pro Klasse. . . . .	215
7.10	Häufig16K-Daten: Wichtigkeit der Variablen. . . . .	217
7.11	SOFA1K-Daten: Korrekte Klassifikationen. . . . .	218
7.12	SOFA1K-Daten: Fehler auf den Trainings- und Testdaten. . . . .	218
7.13	SOFA1K-Daten: Anzahl der erzeugten Regeln pro Klasse. . . . .	218
7.14	SystThrombo2K-Daten: Korrekte Klassifikationen. . . . .	221
7.15	SystThrombo2K-Daten: Fehler auf den Trainings- und Testdaten. . . . .	221
7.16	SystThrombo2K-Daten: Anzahl der erzeugten Regeln pro Klasse. . . . .	221
7.17	Wichtig6K-Daten: Korrekte Klassifikationen. . . . .	223
7.18	Wichtig6K-Daten: Fehler auf den Trainings- und Testdaten. . . . .	223
7.19	Wichtig6K-Daten: Anzahl der erzeugten Regeln pro Klasse. . . . .	224
7.20	SystDiasThrombo3K-Daten: Korrekte Klassifikationen. . . . .	225
7.21	SystDiasThrombo3K-Daten: Fehler auf den Trainings- und Testdaten. . . . .	226
7.22	SystDiasThrombo3K-Daten: Anzahl der erzeugten Regeln pro Klasse. . . . .	227
7.23	Klinikdaten: Repräsentative Regelmengen. . . . .	230
7.24	Klinikdaten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln. . . . .	233
7.25	Klinikdaten: Neu erzeugte, möglich zu erzeugende Itemsets und äußere Generalisierungsindices. . . . .	233
7.26	Klinikdaten: Wichtigkeit der Items. . . . .	234
7.27	Klinikdaten (35 wichtigste Items): Neu erzeugte, möglich zu erzeugende Itemsets und äußere Generalisierungsindices. . . . .	235

7.28	Klinikdaten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln unter Berücksichtigung der Wichtigkeit. . . . .	236
7.29	Klinikdaten: Wichtigkeit der 35 wichtigsten Itemsets. . . . .	237
7.30	SystDiasThrombo3K/Sym35-Daten: Korrekte Klassifikationen. . . . .	244
7.31	SystDiasThrombo3K/Sym35-Daten: Fehler auf den Trainings- und Testdaten. . . . .	245
7.32	SystDiasThrombo3K/Sym35-Daten: Anzahl der erzeugten Regeln pro Klasse. . . . .	245
7.33	Häufig16K-Daten: Plausible Alarme. . . . .	251
7.34	SOFA1K-Daten: Plausible Alarme. . . . .	253
7.35	SystThrombo2K-Daten: Plausible Alarme. . . . .	254
7.36	Wichtig6K-Daten: Plausible Alarme. . . . .	255
7.37	SystDiasThrombo3K-Daten: Plausible Alarme. . . . .	256
A.1	Trenddaten der Vorstudie. . . . .	264
A.2	Zeitstempel der Trenddaten der Vorstudie. . . . .	265
A.3	Variablen der ausgewählten symbolischen Klinikdaten I. . . . .	270
A.4	Variablen der ausgewählten symbolischen Klinikdaten II. . . . .	271
C.1	IRIS-Daten: Konfidenz der Regeln pro Klasse. . . . .	280
C.2	IRIS-Daten: Häufigkeit der Regeln pro Klasse. . . . .	280
C.3	Cancer1-Daten: Konfidenz der Regeln pro Klasse. . . . .	280
C.4	Cancer1-Daten: Häufigkeit der Regeln pro Klasse. . . . .	281
C.5	Diabetes1-Daten: Konfidenz der Regeln pro Klasse. . . . .	281
C.6	Diabetes1-Daten: Häufigkeit der Regeln pro Klasse. . . . .	281
C.7	Spiraldaten ( $nearby = 2$ ): Konfidenz der Regeln pro Klasse. . . . .	281
C.8	Spiraldaten ( $nearby = 2$ ): Häufigkeit der Regeln pro Klasse. . . . .	282
C.9	Spiraldaten ( $nearby = 0.14$ ): Konfidenz der Regeln pro Klasse. . . . .	282
C.10	Spiraldaten ( $nearby = 0.14$ ): Häufigkeit der Regeln pro Klasse. . . . .	282
C.11	Diabetes1-Daten (fehlende Werte): Konfidenz der Regeln pro Klasse. . . . .	282
C.12	Diabetes1-Daten (fehlende Werte): Häufigkeit der Regeln pro Klasse. . . . .	283
C.13	NormAusreißer-Daten: Konfidenz der Regeln pro Klasse, Fuzzy-RecBF-DDA. . . . .	283
C.14	NormAusreißer-Daten: Häufigkeit der Regeln pro Klasse, Fuzzy-RecBF-DDA. . . . .	283
C.15	NormAusreißer-Daten: Konfidenz der Regeln pro Klasse, Fuzzy-Wv-RecBF-DDA. . . . .	283
C.16	NormAusreißer-Daten: Häufigkeit der Regeln pro Klasse, Fuzzy-Wv-RecBF-DDA. . . . .	284
C.17	Häufig12Vor-Daten: Konfidenz der Regeln pro Klasse. . . . .	284
C.18	Häufig12Vor-Daten: Häufigkeit der Regeln pro Klasse. . . . .	284
C.19	Wichtig17Vor-Daten: Konfidenz der Regeln pro Klasse. . . . .	285
C.20	Wichtig17Vor-Daten: Häufigkeit der Regeln pro Klasse. . . . .	285
C.21	Häufig16K-Daten: Konfidenz der Regeln pro Klasse. . . . .	285

---

C.22 Häufig16K-Daten: Häufigkeit der Regeln pro Klasse. . . . .	285
C.23 SOFA1K-Daten: Konfidenz der Regeln pro Klasse. . . . .	286
C.24 SOFA1K-Daten: Häufigkeit der Regeln pro Klasse. . . . .	286
C.25 SystThrombo2K-Daten: Konfidenz der Regeln pro Klasse. . . . .	286
C.26 SystThrombo2K-Daten: Häufigkeit der Regeln pro Klasse. . . . .	286
C.27 Wichtig6K-Daten: Konfidenz der Regeln pro Klasse. . . . .	287
C.28 Wichtig6K-Daten: Häufigkeit der Regeln pro Klasse. . . . .	287
C.29 SystDiasThrombo3K-Daten: Konfidenz der Regeln pro Klasse. . . . .	287
C.30 SystDiasThrombo3K-Daten: Häufigkeit der Regeln pro Klasse. . . . .	287
C.31 Häufig16K-Daten: Alarmkennzahlen. . . . .	288
C.32 SOFA1K-Daten: Alarmkennzahlen. . . . .	288
C.33 SystThrombo2K-Daten: Alarmkennzahlen. . . . .	288
C.34 Wichtig6K-Daten: Alarmkennzahlen. . . . .	289
C.35 SystDiasThrombo3K-Daten: Alarmkennzahlen. . . . .	289



# Symbolverzeichnis

$\emptyset$	Leere Menge
$A \subsetneq B$	$A$ ist echte Teilmenge von $B$
$\uplus$	Disjunkte Mengenvereinigung
$ \cdot $	Betrag
$\ \cdot\ $	Norm
$\forall$	Allquantor
$\exists$	Existenzquantor
$\sharp, \ \ $	Anzahl
$-$	Minus, mengentheoretische Differenz
$\nabla$	Gradientenoperator
$a := b$	$a$ wird durch $b$ definiert
$a \approx b$	$a$ approximativ gleich $b$
$a \gg b$	$a$ bedeutend größer als $b$
$a \ll b$	$a$ bedeutend kleiner als $b$
$(x_1, \dots, x_n; k)$	$n$ -dim. Datentupel aus der Klasse $k$
$X = x$	$x$ ist konkrete Realisierung der Zufallsvariablen $X$
$\bar{x}$	Arithmetisches Mittel

---

$\vec{v}, v$	Vektor $v$ (meistens ohne Pfeil verwendet)
$AssRule$	Menge der gültigen Assoziationsregeln
$conf(I \Rightarrow J)$	Konfidenz der Assoziationsregel $I \Rightarrow J$
$conf_{K_l}(I \Rightarrow K_l)$	Konfidenz der Generalisierungsregel $I \Rightarrow K_l$
$Conf_{K_l}$	Menge der konfidenten Regeln bez. der Klasse $K_l$
$COR(X, Y)$	Korrelation zweier Zufallsvariablen $X, Y$
$\langle xy^T \rangle, \widetilde{COR}(X, Y)$	Empirische Korrelation zweier Zufallsvariablen $X, Y$
$d(\cdot, \cdot)$	Metrik
$\Delta W$	Änderung einer Gewichtsmatrix $W$ , d.h. $\Delta W = W(n+1) - W(n)$
$\mathcal{DS}$	Datensatz
$E(X)$	Erwartungswert einer Zufallsvariablen $X$
$\tilde{E}(X)$	Empirischer Erwartungswert einer Zufallsvariablen $X$
$earliness(R)$	Frühzeitigkeit einer Regel $R$
$f_X$	Wahrscheinlichkeitsdichtefunktion einer Zufallsvariablen $X$
$freq(I)$	Häufigkeit des Itemsets $I$
$freq(I \Rightarrow J)$	Häufigkeit der Assoziationsregel $I \Rightarrow J$
$freq(I \Rightarrow K_l)$	Häufigkeit der Generalisierungsregel $I \Rightarrow K_l$
$freq_{K_l}(I \Rightarrow K_l)$	Abdeckung der Generalisierungsregel $I \Rightarrow K_l$
$Freq_{K_l}$	Menge der häufigen Regeln bez. der Klasse $K_l$

---

$G^{(empty)}$	Leer-Index
$G^{(exist)}$	Existenz-Index
$G^{(in)}$	Innerer Generalisierungsindex
$G^{(out)}$	Äußerer Generalisierungsindex
$GenRule$	Menge der gültigen Generalisierungsregeln
grad	Gradient
$I$	Identity matrix (dt.: Einheitsmatrix), Itemset
$I_1 I_2$	Matrixprodukt, disjunkte Vereinigung zweier Itemsets
$imp_{K_l}(V)$	Wichtigkeit der Variablen $V$ bez. der Klasse $K_l$
$\mathcal{IS}$	Menge der Itemsets
$\mathcal{IS}(I)$	Menge der Itemsets, die $I$ enthalten
$\mathcal{IS}_k$	Menge der Itemsets, die zur Klasse $k$ gehören
$\mathcal{IS}^k$	Menge der generalisierten Itemsets, die im $k$ -ten Level entstehen
$\mathcal{IS}_k(I)$	Menge der Itemsets, die zur Klasse $k$ gehören und die $I$ enthalten
$\mathcal{IS}^F$	Menge aller generalisierten Itemsets
$\mathcal{IS}^{F \geq} = GenRule$	Menge aller generalisierten Itemsets, die die Minimum-Bedingungen erfüllen
$\mathcal{IS}_{notk}$	Menge der Itemsets, die nicht zur Klasse $k$ gehören
$\mathcal{IV}$	Menge der Itemvektoren
$I(x_k)$	Information eines Ereignisses $x_k$

---

$\text{index}(x_i)$	$i := \text{index}(x_i)$ oder $j := \text{index}(\min_i\{x_i\})$
min	Minimum einer Menge
max	Maximum einer Menge
$\mu_A$	Zugehörigkeitsfunktion
$\mathbb{N}$	Menge der natürlichen Zahlen
$\mathbb{N}_n$	Menge der natürlichen Zahlen bis $n$
$\mathcal{O}(\cdot)$	Ordnung
$\mathcal{P}(M)$	Potenzmenge der Menge $M$
$P(X = x)$	Wahrscheinlichkeit für das Ereignis $x$
$P(X = x Y = y)$	Bedingte Wahrscheinlichkeit für das Ereignis $x$ bez. $y$
$\mathbb{R}$	Menge der reellen Zahlen
$\text{sim}(I, J)$	Ähnlichkeit zweier Regeln
$\text{sim}(R, \hat{R})$	Ähnlichkeit zweier Regelmengen
$wconf_{K_l}(I \Rightarrow K_l)$	Gewichtete Konfidenz der Generalisierungsregel $I \Rightarrow K_l$
$x^T y, \langle x, y \rangle$	Kanonisches Skalarprodukt für $x, y \in \mathbb{R}^n$
$x^{(m)}$	Vektor metrischer Werte
$x^{(s)}$	Vektor symbolischer Werte

Die Symbole für die Parameter der RBF-DDA-, RecBF-DDA- und Fuzzy-RecBF-DDA-Netze sind in den Abschnitten 2.2.3 und 3.1 zusammengestellt.

# Abkürzungsverzeichnis

A-Alg.	Assoziationsregel-Algorithmus
ARMA	Autoregressive Moving Average
ASK	Ausgewählte Symbolische Klinikdaten
BSS	Blind Source Separation
DDA	Dynamic Decay Adjustment
dim.	dimensional
EEG	Elektroenzephalogramm
EKG	Elektrokardiogramm
FB	Feedback
FF	Feedforward
G-Alg.	Generalisierungsregel-Algorithmus
(G)CS	(Growing) Cell Structures, Glasgow Coma Score
gew.	gewichtet
(G)NG	(Growing) Neural Gas
ICA	Independent Component Analysis
IDA	Intelligent Data Analysis
Irr.	Irrelevant

Klasse ü	Klasse „überlebt“
Klasse v	Klasse „verstorben“
LICA	Lineare ICA
LVQ	Lernende Vektorquantisierung
MEDAN	Medizinische Datenanalyse mit Neuronalen Netzen
MDS	Multidimensionale Skalierung
MSE	Mean Squared Error
Pat.	Patient(en)
PCA	Principal Component Analysis
PNN	Probabilistisches Neuronales Netz
RBF	Radiale Basisfunktion(en)
RCE	Restricted Coulomb Energy
RecBF	Rechteckige Basisfunktion(en)
ROC	Receiver Operating Characteristic
RR	(mit dem Riva-Rocci-Apparat gemessener) Blutdruck
SOM	Self Organizing Map
SPI	Similarity Placement Index
SVM	Support Vector Machine
WTA	Winner-Takes-All
Wv	Weight Vector

Die Abkürzungen der medizinischen Scores findet der Leser im Abschnitt 2.1.3.

# Kapitel 1

## Einleitung

Ein sehr wichtiges Forschungsgebiet der Intensivmedizin ist die Ursachenforschung des *septischen Schocks*. Der septische Schock ist weltweit eine der häufigsten Todesursachen auf Intensivstationen. Die medizinische Wissenschaft widmete sich bislang vielen Detailfragen zur medizinischen Diagnose und Therapie des septischen Schocks. Für viele Analysen standen aber bislang nur kleine Patientenzahlen und eine kleine Anzahl von Messwerten in einzelnen Kliniken zur Verfügung.

Die zentrale Fragestellung im Projekt *MEDAN* (Abk. für: Medizinische Datenanalyse mit Neuronalen Netzen) lautet: „Warum führt der septische Schock zum Tod bei einer hohen Anzahl von Patienten und warum überleben wiederum andere?“- Die generelle Hypothese hierzu lautet: „Der Einsatz adaptiver Datenanalyseverfahren liefert neue, bislang unbekannte Erkenntnisse über den septischen Schock.“ Die Hypothese beschränkt sich dabei bewusst auf die Verwendung der z.Z. routinemäßig in Patientenakten dokumentierten medizinischen Daten und beinhaltet keine aufwendigen, teuren Messungen von biochemischen Substanzen, die durch Immunsystemreaktionen freigesetzt werden. Die medizinische Bedeutung des septischen Schocks wird dem Leser im Abschnitt 5.1 dargelegt.

Was hat aber medizinische Datenanalyse mit Informatik zu tun? Ein Ziel im Projekt *MEDAN* ist der Aufbau einer Datenbank, in der alle Daten der Patientenakten von septischen Schock-Patienten abgespeichert werden, um sie für computerunterstützte Analysen verwendbar zu machen, aber auch um Ärzten via Internet Abfragen auf diese Datenbank zu ermöglichen. Also ist die Erstellung und das Management der Datenbank Aufgabe der (Medizinischen) Informatik.

Das Anfertigen von Analysen per Hand nach Auszählen einiger weniger Daten von einer kleinen Anzahl handgeschriebener Patientenakten führt i.d.R. nicht zu statistisch signifikanten Aussagen. *Wissenschaftliche* Datenanalyse sollte aktuelle Datenanalyseverfahren verwenden, die Massendaten automatisch auswerten können. Das bedeutet: Die Bereitstellung von Methoden zur automatischen, intelligenten Massendatenauswertung ist eine Aufgabe für Informatiker. Die Datenanalyse erfordert

oft eine gute mathematische Vorbildung, so dass hier eine interdisziplinäre Zusammenarbeit mit Mathematikern bzw. Statistikern sinnvoll ist.

Im Rahmen des Projekts entstand die Idee zu dieser Dissertation. Sie beschreibt und verwendet adaptive Datenanalyseverfahren und beinhaltet als Schwerpunkte sowohl die Weiterentwicklung und Neukonzeption von Verfahren zur metrischen und symbolischen Regelgenerierung mit integrierter Klassifikation zur Diagnose als auch deren Anwendung auf die medizinischen septischer Schock-Daten aus der o.g. Datenbank und zwei weiteren medizinischen Datenbanken. Die Anwendungsergebnisse in den Kapiteln 5, 6 und 7 sind zugleich die Anwendungsergebnisse des Projekts. Durch die Arbeit in den Kapiteln 2 bis 4 wurden die Anwendungsergebnisse in dieser Form ermöglicht.

Anhand der vorhandenen Datenbanken können viele verschiedene Analysen durchgeführt werden: die *Medizinische Datenanalyse* ist ein sehr interessantes und umfangreiches wissenschaftliches Betätigungsfeld. Vielfältige Modelle und Verfahren werden auf die verschiedensten Probleme angewendet. - Keinesfalls sollen in dieser Schrift alle bisherigen Versuche zahlreicher Wissenschaftler dokumentiert werden, die einmal auf diesem Gebiet gearbeitet haben. Vielmehr soll anhand des bedeutenden medizinischen Problems *septischer Schock* und anhand weiterer Benchmarkdaten die Anwendbarkeit der *adaptiven Datenanalyse* aufgezeigt werden, deren Bauteile der Informatik, der Mathematik, der Physik, der Statistik, den Ingenieurwissenschaften und der Medizin entstammen, so dass sich ein hoher Grad an Interdisziplinarität ergibt.

Diese Arbeit beurteilt hauptsächlich die Anwendbarkeit der adaptiven Regelgenerierung auf die Problematik des septischen Schocks und untersucht zusätzlich einige grundsätzliche Fragestellungen der adaptiven Regelgenerierung. Regeln unterstützen den Anwender, eine Diagnoseentscheidung durch ein Klassifikationsverfahren explizit zu verstehen. Das Verstehen von Systementscheidungen ist absolut notwendig, wenn Therapieentscheidungen evaluiert werden sollen.

Wie bereits erwähnt, handelt es sich bei dem Forschungsgebiet *Datenanalyse* um ein enorm großes und vielfältiges, das von vielen Disziplinen beeinflusst wird und Anwendungen in der Medizin, Physik, Biologie, Chemie, Geo- und Umweltwissenschaften, Technik, Betriebs- und Volkswirtschaft, Psychologie, Soziologie u.v.m. hat. Viele Methoden wurden und werden für die verschiedensten Probleme entworfen bzw. verfeinert.

Umfangreich erweitert wird die klassische Statistik um Verfahren aus den Bereichen *Data Mining*, *Intelligent Data Analysis*, *Knowledge Discovery* und *Machine Learning*; zu diesen Themen werden im Verlauf der Arbeit zahlreiche Literaturhinweise gegeben.

---

Hervorheben möchte ich allerdings im Folgenden die *Adaptivität* von Verfahren, d.h. die Anpassung der Verfahren an die Daten. Der Begriff *Adaptivität* ist allgemein und auch problemorientiert gefasst; somit ist er für die Datenanalyse geeignet und umfasst insbesondere die Verfahren aus den o.g. Gebieten. Gemeint ist die Generierung von Wissen aus den Daten, sei es probabilistisch, statistisch, iterativ oder direkt berechnend, neuronal usw. oder auch kombiniert. Bewusst wird auf eine strikte Einhaltung von blockierenden Denkgrenzen verzichtet, wenngleich klar ist, dass die Beherrschung aller Details dieser Gebiete unmöglich ist. Insbesondere *kompliziert strukturierte* umfangreiche Datenmengen können von adaptiven Datenanalysen profitieren, da man hier i.d.R. keine einfachen Modellannahmen für die Daten zur Hand hat, die hilfreich eingesetzt werden können. Das ist auch im Fall der septischer Schock-Daten so.

Nachdem man erkannt hat, dass einem Arzt Grenzen gesetzt sind, Entscheidungen (z.B. Diagnosen) allein auf der Grundlage seiner Erfahrung zu treffen, d.h. ohne Verwendung von datenbasiert erstellten mathematisch-technischen Systemen, gelangt man zu der Erkenntnis, dass der Einsatz adaptiver Verfahren für die medizinische Datenanalyse sehr vorteilhaft ist. Dieser Standpunkt wird in [Bra99] einführend diskutiert. Auf der Grundlage [Bra99] wird in dieser Arbeit gezeigt, wie man vorhandene, umfangreiche Datenbanken zur adaptiven Datenanalyse und speziell zur adaptiven Regelgenerierung verwenden kann.

An dieser Stelle möchte ich dem interessierten Leser einen Überblick über den Aufbau und den Inhalt der Arbeit geben. Dabei möchte ich ausdrücklich auf die hohe Komplexität des medizinischen Datenmaterials hinweisen (multivariate Zeitreihen mit Messungen in unregelmäßigen, unterschiedlichen Zeitabständen mit einer großen Menge an fehlenden Werten, die durch Messungen am Patienten entstanden sind). Hinzu kommen sog. Stammdaten ohne zeitliche Dimension wie z.B. Geschlecht und symbolische Daten wie Operationen und Diagnosen. Weiterhin möchte ich auch ausdrücklich auf die Schwierigkeit hinweisen, für derart komplexes, umfangreiches, fehlerbehaftetes Datenmaterial Verfahren auszuwählen bzw. zu entwickeln, für die man eine Verwendbarkeit vermutet. Dieser Situation müssen die eingesetzten adaptiven Verfahren standhalten.

Im Kap. 2 werde ich grundlegend die Ideen, Funktionsweisen und Anwendungsgebiete von *adaptiven* Verfahren anhand der Literatur vorstellen. Es werden neben aktuellen Verfahren auch die Standardmethoden genannt. Im 2. Kapitel soll insbesondere, als Vorbereitung für die weiteren Kapitel, auf die Grundlagen der medizinischen Standardanalysemethoden (z.B. *Scores*), der *Datenanalyse*, der *neuronalen Netze*, der *Klassifikation* und der *Regelgewinnung* eingegangen werden. Auch wird die grundsätzliche Bedeutung der vorgestellten Datenanalyseverfahren für die Anwendung auf die septischer Schock-Daten bewertet.

Die eigenen Arbeiten in den Kapiteln 3 bis 7 umfassen im Wesentlichen folgende Themenbereiche:

- Verbesserung eines Neuro-Fuzzy-Verfahrens zur Regelgewinnung und Klassifikation durch Verbesserung des Überlappungsverhaltens der Regeln (Abschnitt 3.1 und 3.6) für die uneingeschränkte Anwendung auf metrischen Realweltdaten,
- Untersuchung der Erzeugung rechteckiger, scharfer Regeln aus metrischen Daten durch das Neuro-Fuzzy-Verfahren (Abschnitt 3.4), falls wie im Fall der metrischen septischer Schock-Daten keine sinnvolle Interpretation linguistischer Fuzzy-Variablen möglich ist,
- Entwicklung eines Verfahrens zur Erzeugung von durchschnittsbasierten Generalisierungsregeln aus symbolischen Daten (Abschnitt 4.1), das grundsätzlich sinnvolle Regeln für das medizinische Anwendungsproblem erzeugen kann, d.h. das mit mehrfachen, nicht-eindeutigen Klassenzugehörigkeiten, fehlenden Werten und Negationen von Attributen umgehen kann,
- Definition einer „gewichteten Konfidenz“ zur Klassifikation auf der Basis der Generalisierungsregeln, die unterschiedliche Klassenhäufigkeiten berücksichtigt (Abschnitte 2.5.2 und 4.3),
- Diskussion der Eigenschaften der Generalisierungsregeln, z.B. Leveleigenschaft, Optimalität der Regeln, Verwendung von Negationen, fehlende Werte (Abschnitt 4.2),
- Heuristiken zur Laufzeitverbesserung der Generalisierung, da eine vollständige Regelgenerierung i. Allg. zu aufwendig ist und zu viele Regeln erzeugt (Abschnitte 4.6, 4.7),
- grundsätzlicher und empirischer Vergleich zu Assoziationsregeln und FP-Bäumen (Abschnitte 4.8 und 6.6), der die Sinnhaftigkeit der Generalisierungsregeln belegt,
- Anwendung der Verfahren auf Benchmark-Daten zur Beurteilung der grundsätzlichen Leistungsfähigkeit der entwickelten Verfahren, d.h. Bewertung der Güte der Regeln und der Klassifikation (Abschnitte 3.2, 3.6 und 4.4),
- Anwendung der verschiedenen Verfahren zur Regelgenerierung und Klassifikation auf die septischer Schock-Daten, inkl. der Diskussion der notwendigen Datenvorverarbeitungen (Abschnitte 3.3, 5.5, 6.2 bis 6.6, 7.2 und 7.3), Vergleich der metrischen automatischen Regelgewinnung mit der manuellen Regelerzeugung (Abschnitt 5.5.2),

- Kombination der symbolischen und metrischen Regelgewinnung, um zu untersuchen, ob symbolische und/oder metrische Regeln besser zur Klassifikation und insbesondere zum Bau eines Alarmsystems geeignet sind (Abschnitte 2.5.3 und 7.4),
- Erstellung ergänzender deskriptiver Standardstatistiken (u.a. Korrelationen), insbesondere interessant zur Beurteilung der Datenqualität, z.B. durch Beeinträchtigung fehlender Werte (Abschnitte 5.2 bis 5.4, 6.1, 7.1),
- Einführung und Anwendung einer „Wichtigkeit“ für Variablen zur Merkmalsauswahl (für das Neuro-Fuzzy-Verfahren im Abschnitt 3.5 und für die Generalisierungsregel-Methode im Abschnitt 4.5), da weniger Merkmale zu verständlicheren Regeln und einer besseren Klassifikationsleistung führen können,
- Definition der „Frühzeitigkeit“ von Regeln (Abschnitte 3.5 und 4.5), da Regeln, die evtl. früher als andere auftreten, interessanter für ein Frühwarnsystem sind,
- Entwicklung eines Kriteriums für die Modellwahl auf der Basis der Ähnlichkeit von Neuro-Fuzzy-Regeln (Abschnitt 3.7) bzw. von Generalisierungsregeln (Abschnitt 4.9), um ein konkretes Modell aus einer Menge von verschiedenen Modellen aus verschiedenen Versuchswiederholungen in der Anwendung einsetzen zu können, aber auch um dem Anwender eine Menge unähnlicher Regeln vorzustellen, ohne ihm redundante Regeln zu präsentieren,
- Entwicklung eines Alarmsystems für septische Schock-Patienten (Abschnitt 7.5): Für die Anwender ist der Höhepunkt der Datenanalyse sicher die Einführung eines Alarmsystems auf der Basis der septischer Schock-Daten; ein Großteil der bisher genannten Analysemethoden wird hierfür benötigt.

Ein zusammenfassender Ausblick folgt im 8. Kapitel. Die verwendeten Daten werden im Anhang A beschrieben. Dort wird auch ein kurzer Überblick über die erstellte und verwendete Software gegeben. Der Anhang B enthält die ROC-Kurven der Klassifikationsergebnisse; der Anhang C beinhaltet zusätzliche Ergebnistabellen in Ergänzung der Abschnitte 3.2 bis 3.4, 3.6, 5.5, 7.2.2 und 7.5. Im Anhang D ist der Algorithmus [HB95] (Fuzzy-RecBF-DDA) wiedergegeben.

Im Internet existieren mittlerweile fast zu jedem Stichpunkt, Verfahren oder Autor Web-Seiten mit zahlreichen Informationen. Leider wechseln die Web-Adressen häufig, so dass immer die Quelle der Papierversion angegeben wird und auf eine nicht dauerhafte Angabe von Internet-Seiten im Literaturverzeichnis verzichtet wird. Es erscheint lediglich zusätzlich die Angabe *Internet*. Mit Hilfe gängiger Internet-Suchmaschinen findet der Leser hoffentlich schnell die entsprechenden *noch vorhandenen* Seiten.

Die in der medizinischen Fachliteratur auftretenden Fachbegriffe können im medizinischen Wörterbuch [Psc94] nachgeschlagen werden. Sie werden im Rahmen dieser Arbeit im Text nur knapp erläutert.

Wird im Verlauf der Arbeit ein Absatz in Kleindruck gesetzt, so stellt der Absatz eine zusätzliche Information für den Leser bereit, die aber nicht zum „roten Faden“ der Arbeit gehört.

# Kapitel 2

## Adaptive Datenanalyse in der Medizin

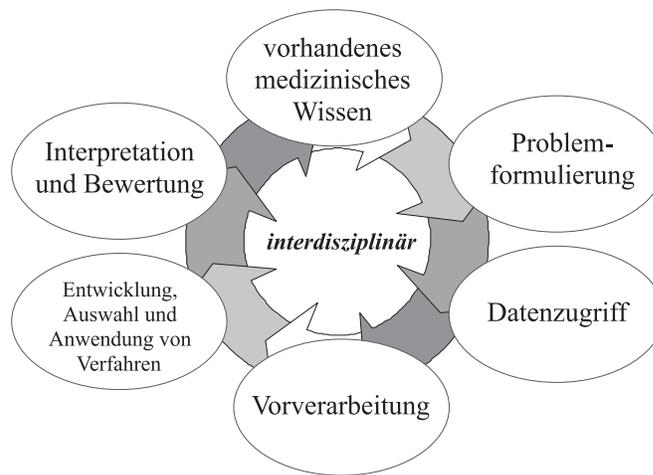
In diesem Kapitel werden adaptive Datenanalyseverfahren vorgestellt, die von grundsätzlicher Bedeutung für den weiteren Fortgang (Kapitel 3 bis 7) sind. **Adaptiv** soll heißen: Das jeweilige Verfahren passt sich seiner Umgebung (hier den Daten) an. Sinnvoll kann eine solche Anpassung sein, wenn eine exakte Berechnung einer Lösung theoretisch durchführbar ist, aber z.B. aufgrund der benötigten Zeitkomplexität nicht möglich ist oder aber eine optimale theoretische Lösung nicht existiert. In diesem Fall können *adaptive Verfahren* helfen, suboptimale Lösungen mit vertretbarem Zeitaufwand zu finden. Unter einem **adaptiven Verfahren** verstehe ich im Folgenden Verfahren, die von vorliegenden Daten ausgehend eine (suboptimale) Lösung mit vertretbarem Zeitaufwand auffinden.

Beispielhaft werden Anwendungen der verschiedenen Verfahren genannt, und es wird die Verwendbarkeit der jeweiligen Verfahren für den Fortgang dieser Arbeit, d.h. für die Analyse der septischer Schock-Patientendaten, diskutiert. Besonders interessant kann das 2. Kapitel, das größtenteils einen eher einleitenden Charakter hat, für alle diejenigen sein, die sich in Zukunft verstärkt mit adaptiven Methoden in der Medizin oder einem anderen Anwendungsgebiet beschäftigen möchten. - Beginnen werde ich das Kapitel mit der Vorstellung der Besonderheiten der *medizinischen* Datenanalyse.

### 2.1 Datenanalyse in der Medizin

*Datenanalyse in der Medizin* gliedert sich ausgehend vom vorhandenen medizinischen Wissen in folgende Arbeitsschritte (Abb. 2.1), die nach eigener Erfahrung am besten auf den aufgeführten Personenkreis (M = Mediziner, S = Software Entwickler, D = Datenanalyst) aufgeteilt werden können:

- Problemformulierung (M und D),
- Datenverwaltung (S),



**Abbildung 2.1:** Datenanalyse-Prozess in der Medizin.

- Vorverarbeitung (S und D),
- Entwicklung, Auswahl und Anwendung von Verfahren (D) sowie
- Interpretation und Bewertung (M und D).

Dabei ist bemerkenswert, dass ein großer Arbeitsanteil für den Mediziner selbst entsteht, da zu jedem Zeitpunkt das Anwenderwissen eine große Rolle spielt. Auch ist ein großer Programmieraufwand erforderlich, so dass die Durchführung eines medizinischen Datenanalyseprojekts keinesfalls die alleinige Arbeit des Datenanalysten ist, sondern eine interdisziplinäre Arbeit mit Programmierunterstützung. Allerdings hat der Datenanalyst die Aufgabe, einen Gesamtüberblick zu bewahren.

Trotz der Anwendung *moderner* Verfahren, z.B. aus dem Bereich des *Data Mining* spielen auch *bewährte* Methoden in der Medizin eine große Rolle. Aus diesem Grund werden in den nächsten Unterabschnitten auch diese Methoden besprochen. Ziel aller Analysen ist der Erkenntnisgewinn zugunsten des Patienten.

Verwandte bzw. neuere Begriffe für *Datenanalyse*, die den Bereich der klassischen Statistik erweitern bzw. verlassen, sind *Data Mining* [HK00], *Knowledge Discovery in Databases* (Abk.: KDD, <http://www.kdnuggets.com>, Stand: Dez. 2001), *Machine Learning* ([Mit97], [Mit99], dt.: Maschinelles Lernen) und *Intelligent Data Analysis* (Abk.: IDA, [BH99] oder [Run00]). Im Folgenden wird aber zweckmäßig und sinn gemäß weiter der Begriff *Datenanalyse* verwendet. Eine sinnvolle Unterscheidung wird z.Z. zwischen *kommerzieller* und *wissenschaftlicher* Datenanalyse getroffen. Die kommerzielle Datenanalyse ist profitorientiert und widmet sich weitestgehend der Anwendung bekannter Verfahren (Warenkorbanalyse, „Direct-Mailing-Aktionen“ als Marketinginstrument, siehe [KWZ98]). Die wissenschaftliche Datenanalyse ist nicht

profitorientiert (z.B. Auswertung medizinischer oder astronomischer Daten) und versucht die Methoden für Anwendungen zu verbessern und grundlegende Eigenschaften der Analysemethodik herauszuarbeiten.

Bei Interesse an medizinischer Datenanalyse werden begleitende Werke wie [vBM97] (Einführung in die *Medizinische Informatik*), [Har98] (Einführung in die *Statistik für Mediziner*) und [See97] (Mischung aus Analyse und medizinischer Informatik unter Berücksichtigung von neuronalen Netzen und Modellbildung) empfohlen.

Grundlage einer medizinischen Datenanalyse ist eine Datenbank, die die medizinischen Daten enthält (Abschnitt 2.1.1). Einfache Analysen wie die Berechnung von Kennzahlen oder die Erstellung von Histogrammen dienen dem Arzt oder dem Datenanalysten dazu, sich schnell einen Überblick über die Daten zu verschaffen (Abschnitt 2.1.2). Eine besonders beliebte Methode der medizinischen Datenanalyse ist die Berechnung von sog. Scores (Abschnitt 2.1.3). Die Erstellung einer medizinischen Datenbank und deren Auswertung kann Bestandteil einer medizinischen Studie sein (Abschnitt 2.1.4). Im Rahmen des MEDAN-Projekts ist insbesondere die Erstellung eines Alarmsystems interessant; die grundlegende Funktionsweise eines Alarmsystems wird deshalb im Abschnitt 2.1.5 beschrieben.

### 2.1.1 Medizinische Datenbasen

Im Krankenhaus werden für jeden Patienten Daten auf Papier (*Krankenblatt* oder *Patientenkurve* genannt) niedergeschrieben, siehe z.B. [vBM97, Kap. 7]. Dokumentiert werden i.d.R. folgende Daten: Aufnahmedaten (z.B. Name, Geschlecht, Alter) bei Einlieferung in das Krankenhaus, Vorgeschichte des Patienten (Anamnese, z.B. frühere Operationen), täglich mehrmals oder regelmäßig gemessene Routinewerte wie Blutdruck (Kreislaufdaten) oder Medikamentengaben. Hinzu kommen je nach Fall mikrobiologische Befunde, Röntgenbefunde, Beatmungsprotokolle, usw. – Die für den septischen Schock relevanten und verfügbaren Daten (Vorstudien-, ASK-Daten, Klinikdaten) sind im Anhang A beschrieben.

Bei einem schwer kranken Patienten können so im Laufe von einigen Wochen hunderte Seiten Papier anfallen, so dass die Dokumentation eines Patienten schnell unübersichtlich wird. Einige Krankenhäuser haben deshalb bereits elektronische Dokumentationssysteme. Dort stehen die Daten gemäß den jeweiligen Arztwünschen vor Ort elektronisch bereit, allerdings ist die *Standardisierung* der Daten selten gewährleistet. Für das Projekt MEDAN wurden die Klinikdaten von Doktoranden der Medizin mit großem Aufwand eingegeben; nur für einzelne Kliniken war die Übernahme elektronisch verfügbarer Daten möglich, wengleich aufgrund uneinheitlicher Schnittstellen sich dieser Vorgang nicht unproblematisch dargestellt hat.

### 2.1.1 Vereinbarung: (Datenanalyse in der Medizin)

Unter *Datenanalyse in der Medizin* verstehe ich die *quantitative* Analyse der Daten, initiiert durch Vermutungen, die *daten-* oder *erfahrungsbasiert* aufgestellt werden können unter Verwendung der Patientenkurven innerhalb eines grundsätzlich abgesteckten Problemkreises.

*Data Mining* betont vorrangig die Analyse von Daten *ohne* erfahrungsbasiertes Vorwissen auf der Basis sehr großer Datenbanken. Hierzu findet man in [AZ96] und [KWZ98] Beispiele, auch für nicht-medizinische Anwendungsfelder. In der Denkweise von Betriebswirten dient Data Mining der Steigerung des Unternehmenserfolgs. In Fall des septischen Schocks wäre dann der „Unternehmenserfolg“ die Senkung der Letalität des septischen Schocks auf der Intensivstation. [BM99], [Lav99], [HM01] und [AHL98] geben Überblicke über medizinisches Data Mining, d.h über Wissensextraktion aus medizinischen Datenbanken. [Sch00, Kap. 9] extrahiert durch Benutzung des Programms „IBM Intelligent Miner for Data“ aus medizinischen Datenbanken potenzielles Anwenderwissen. Der Vorteil eines in der Praxis nie in reiner Form durchgeführten Data Minings wäre die Unvoreingenommenheit bei der Analyse im Vergleich zum Anwendungsexperten, der schwerwiegende Nachteil wäre das Ignorieren von wertvollen Zusatzinformationen. Unter dem Sammelbegriff *Evidence Based Medicine*<sup>1</sup> verbirgt sich umgekehrt die Forderung an den Arzt, abgesicherte Datenanalyse-Ergebnisse zu akzeptieren und in die medizinische Praxis umzusetzen. – Im Gegensatz zum Auffinden von Datenanalysewissen aus den Daten verfolgen reine medizinische *Expertensysteme* das Ziel, dem Arzt *bekannte* Informationen zu vermitteln, siehe z.B. [Hel91, Kap. 7] und [Pau95]. Die in diesem Zusammenhang für den Arzt sehr wichtige *ethische* Bewertung des Einsatzes von Expertensystemen wird in [GS97] diskutiert.

Datenanalysen sind natürlich nicht auf medizinische Anwendungen beschränkt, sondern in allen Anwendungsbereichen beheimatet, z.B. in der Gewässeranalyse [Dap98], der Aufdeckung von Kreditkartenmissbrauch [BLH99], der Lawinenprognose [Sch95], der Klimaanalyse [DWS96] und der Supermarkt-Verkaufsanalyse [Thi98].

Hat man gute Ideen und entsprechende Analysewerkzeuge zur Hand, so heißt das aber noch lange nicht, dass man sinnvolle Analysen durchführen kann! Gerade für die medizinische Analyse spielt die Dokumentationsgenauigkeit und -häufigkeit und die Datenvorverarbeitung eine sehr wichtige Rolle (siehe [PHT00], [Tsu00b], die Abschnitte 5.2 und 7.1 sowie [Wit00, Kap. 3]).

### 2.1.2 Elementare medizinische Standardanalysen

Gern verwendete Analysemethoden -bzw. grafiken in der Medizin sind Histogramme, Boxplots, Überlebenskurven und ROC-Kurven (Abk. für: Receiver-operating-

---

<sup>1</sup>Material findet man hierzu unter <http://www.ebmny.org/> (Stand: Dez. 2001).

characteristic), die die *Sensitivität* und die *Spezitivität* eines Verfahrens, z.B. eines Klassifikators oder eines Scores, gegenüberstellen.<sup>2</sup> Wichtige, grundlegende Bewertungskriterien sind *Spezifität*, *Sensitivität*, *Mortalität* und *Letalität*. Sie sind in [Har98, S. 277] zusammengestellt. Es handelt sich um einfache Verhältniszahlen.

Bevor man Daten mit Datenanalyseverfahren auswertet, muss man sich Gedanken über eine *Skalierung* der Daten machen: Die Wertebereiche der Daten sollten so ähnlich sein wie möglich, um ungewünschte Effekte zu vermeiden, z.B. ist es (numerisch) nicht günstig, Werte im  $10^{-3}$ -Bereich mit Werten im  $10^3$ -Bereich zusammen zu verarbeiten. – Je nach Anwendung müssen Messwerte, Medikamente und Diagnosen, die nominale, ordinale oder metrische Variablen sein können, skaliert werden, siehe [HE95, Kap. V].

### 2.1.3 Score-Systeme

*Scores* werden aufgrund der starken medizinischen Verbreitung und ihrer Bedeutung für die Intensivmedizin etwas ausführlicher erläutert.

#### 2.1.2 Definition: (Score)

Ein medizinischer **Score** ist eine Bewertungszahl, die als Indikator für die Schwere einer Krankheit dienen soll.

Wichtige Scores in der Intensivmedizin werden im folgenden Beispiel erklärt. Die Scores a) bis d) aus dem folgenden Beispiel werden im Kap. 7 zur Datenklassifikation verwendet; die Ergebnisse werden dort mit den Ergebnissen verglichen, die mit neuronalen Netzen erzielt werden können.

#### 2.1.3 Beispiel: (Scores)

a) APACHE II (Abk.: für Acute Physiological and Chronic Health Evaluation), [KDWZ85] und [Psc94]: APACHE II ist ein Klassifikationssystem zur Prognoseabschätzung bei Intensivpatienten unter Berücksichtigung von akuten Funktionsstörungen, Alter und vorbestehendem Gesundheitszustand. Der APACHE II-Score bewertet einen Patienten mit 0 bis 71 Punkten. Mit Hilfe des APACHE II-Scores kann man eine Prognose der Überlebenschancen berechnen.

b) SOFA (Abk. für: Sepsis-Related Organ Failure Assessment), [V<sup>+</sup>96], [V<sup>+</sup>98] und [H<sup>+</sup>00]: Der SOFA-Score bewertet Organzustände mit den Werten 1 bis 4. Die Summe dieser Werte wird als SOFA-Score bezeichnet.

c) SAPS II (Abk. für: Simplified Acute Physiology Score), [UK97]: SAPS II ist ein reduzierter APACHE II-Score (statt 34 nur 13 Variablen); SAPS II ermöglicht die Berechnung eines Letalitätsrisikos.

---

<sup>2</sup>Werden in dieser Arbeit Flächeninhalte für ROC-Kurven numerisch berechnet, so wird die Trapezregel verwendet.

d) MODS (Abk. für: Multiple Organ Dysfunction Score), [M<sup>+</sup>95]: MODS ist ein Score, der anhand entsprechender Messwerte Organzustände (Lunge, Leber, Niere, Blutbild, Herz, Neurologie) mit Punkten jeweils von 0 bis 4 bewertet.

e) GCS (Abk. für: Glasgow Coma Score), [Psc94]: Der GCS bewertet Bewusstseinsstörungen (Augenbewegung, Motorik, Sprache) mit Punkten (total 3 Punkte: nicht bei Bewusstsein, 15 Punkte: bei vollem Bewusstsein, 4-14 Punkte: Zwischenstufen).

f) CASA (Abk. für: Clinical Algorithm Structural Analysis), [E<sup>+</sup>97]: Die Behandlung eines Patienten wird in einem Flussdiagramm (Ablaufschema der Behandlung) dargestellt. Verschiedene Aktionen (z.B. Therapien) werden mit verschiedenen Punkten bewertet und aufsummiert.

g) TISS-28 (Abk. für: Therapeutic Intervention Scoring System), [Psc94]: TISS-28 ist ein Bewertungssystem zur Abschätzung des Krankheitsverlaufs und der Prognose bei intensivmedizinischer Behandlung anhand der erforderlichen therapeutischen Maßnahmen.

Die Meinungen über Scores gehen auseinander: Für viele sind sie sehr nützlich, für andere sind sie nicht präzise genug. Zwei Stellungnahmen über Scores werden wiedergegeben. – In [UK97] werden prognostische Scores bewertet und verglichen. Eine Hauptaussage ist: „Probabilistische Scores sind für die individuelle Prognoseabschätzung ungeeignet.“[UK97, S. 476]. Scores liefern gute statistische Aussagen für eine *Gruppe* von Patienten; die *individuelle* Prognosefähigkeit wird auch in [LKT95] bezweifelt, da unterschiedliche Scores unterschiedliche individuelle Prognosen liefern, siehe auch [RF94]. – In [NL96] wird aus diesen Gründen für eine Weiterentwicklung von Scores plädiert, u.a. auch mit Hilfe neuronaler Netze.

### 2.1.4 Studie - Multizentrenstudie

*Studien* werden durchgeführt, um (medizinische) Daten zu erheben, auszuwerten und als Ziel einen (medizinischen) Nutzen aus den Analysen zu ziehen, so auch im MEDAN-Projekt. Unter diesem Gesichtspunkt können die in den vorhergehenden Abschnitt genannten Analysen Teil einer Studie sein. Eine komplizierte Aufgabe im MEDAN-Projekt ist der Aufbau eines Alarmsystems für septische Schock-Patienten (vgl. den nächsten Abschnitt).

Grundlegend im Zusammenhang mit der Erfolgsmessung einer Studie ist der Begriff der *Hypothese*. Eine Hypothese ([HE95, S. 133-137] oder [Bos92, Kap. 4]) ist eine (statistisch formulierte) Vermutung über einen medizinischen Sachverhalt. Um Hypothesen zu belegen oder zu widerlegen, werden in der medizinischen Statistik Test- und Schätzverfahren (z.B. *T*-Test für Gleichheit von Mittelwerten) eingesetzt, siehe

z.B. [Har98, Kap. 7]. Wichtig ist die Angabe eines *Konfidenzbereichs* als Güteabschätzung, in der Medizin i.d.R. zum Niveau 0.05 oder 0.01.

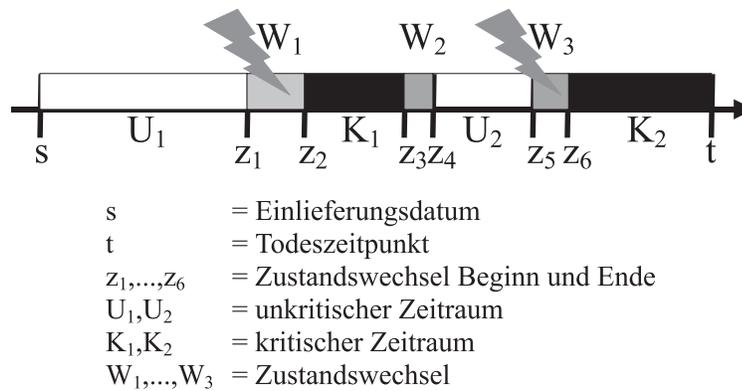
Eine medizinische *Studie* wird also durchgeführt, um die Hypothese zu belegen, dass ein eingesetztes Verfahren oder eine Therapie den Behandlungserfolg steigert. Die Hypothese der Studie im Rahmen des MEDAN-Projekts lautet: Der Einsatz neuronaler Netze<sup>3</sup> kann helfen, die Letalität septischer Schock-Patienten zu senken. Das grundsätzliche Vorgehen und die Probleme bei der Durchführung einer Studie werden in [Har98, Kap. 6] beschrieben. Bei einer *retrospektiven* Studie werden bereits in der Vergangenheit gesammelte Daten analysiert, während in einer *prospektiven* Studie vorausschauend geplant werden kann, welche Daten benötigt werden. Die retrospektive Studie hat den Nachteil, dass das bereits gesammelte Datenmaterial aufbereitet werden muss und bei Fehlern kaum eine Möglichkeit besteht, korrigierend einzugreifen. Die MEDAN-Datenanalysen finden retrospektiv statt, da das z.Z. in den Kliniken erhobene Datenmaterial so unter realistischen Dokumentationsbedingungen ausgewertet werden kann. Allerdings ist eine prospektive Studie geplant, die den klinischen Einsatz der entwickelten Verfahren zusammen mit weiteren Therapievorschlägen von medizinischer Seite rechtfertigen soll.

Um genügend Datenmaterial zu erhalten, wird eine Studie durchgeführt, an der mehrere Kliniken teilnehmen, eine sog. *Multizentrenstudie*. Dies ist z.B. bei Krankheiten erforderlich, für die in einem Krankenhaus nur wenige Patientenfälle ausgewertet werden können, um die Qualität einer Analyse zu erhöhen. Da die septischen Schock-Patienten mit abdominaler Ursache in einzelnen Kliniken selten sind, ist im MEDAN-Projekt eine Multizentrenstudie geplant. Da im MEDAN-Projekt erst ab Beginn der Multizentrenstudie (voraussichtlich ab April 2002) Daten zur Überprüfung der Hypothese gesammelt werden, ist es sinnvoll die Studie mit einem *sequentiellen Design* durchzuführen [BSW86], d.h. eine Maßzahl des Studienerfolges  $Z$  wird ständig statistisch berechnet (und ist von der Patientenzahl abhängig). Je nachdem wie groß der Wert  $Z$  ist, kann eine sequentielle Studie im laufenden Verfahren bei vorzeitigem deutlichen Misserfolg oder Erfolg abgebrochen werden.

Die MEDAN-Studie ist allerdings nicht Inhalt dieser Arbeit. Die Ausführungen sollen nur erläutern, in welchem Kontext die bislang analysierten Daten stehen: Sie dienen dazu, ein Alarmsystem für die septischen Schock-Patienten zu entwerfen (siehe den nächsten Abschnitt). Die quantitativen Bewertungen, die in dieser Arbeit durchgeführten Analysen, werden unabhängig von der Studiendurchführung gegeben.

---

<sup>3</sup>Im Rahmen dieser Arbeit werden allgemeiner *adaptive* Verfahren betrachtet.



**Abbildung 2.2:** Abstrakte Zustände innerhalb einer Patientenkurve. Warnungen („Blitze“) sollten bei einem Alarmsystem so früh wie möglich erfolgen.

### 2.1.5 Alarmsysteme

Fasst man Patientendaten als Zeitreihen auf, so kann man Systeme konstruieren, die bei kritischen Zuständen zu einem Zeitpunkt *Alarm* schlagen. Ein möglicher Alarm ist z.B. der Übergang vom Zustand „normal“ in den Zustand „kritisch“. Dabei darf die *Fehlalarmrate* nicht zu hoch sein. - Zwei Beispiele von existierenden Alarmsystemen sind [Dau98] (Erkennung von Atemstörungen anhand sog. *Changepoints*, das sind abrupte Änderungen in Zeitreihen) und [B<sup>+</sup>97] (Erkennung kritischer Patientenzustände während operativen Eingriffen).

Abrupte Änderungen der gemessenen Variablen sind für das Kollektiv der septischen Schock-Patienten eher zu einem sehr späten Zeitpunkt zu vermuten, z.B. bei einem Multiorganversagen kurz vor dem Tod. Das Auftreten des Multiorganversagens ist für eine Frühwarnung und eine präventive Behandlung während des septischen Schocks nicht mehr sehr nützlich. Ein medizinischer Indikator für eine höhere Letalität ist z.B. die Thrombozytenzahl [V<sup>+</sup>00], deren Wert sich aber i.Allg. nicht abrupt ändert. Eine Vermutung der Ärzte, die zunächst für eine Verwendung von neuronalen Netzen auf den Vorstudienaten der septischen Schock-Patienten *ohne* explizite Beachtung der Zeitreihenstruktur spricht, ist die vermutete bereits eingetretene Veränderung der Variablenwerte zum Zeitpunkt der Einlieferung auf die Intensivstation, d.h. es wäre ein Patient vom Zeitpunkt  $s$  ab entweder eher in einem kritischen Zustand  $K$  oder in einem normalen Zustand  $U$ , siehe Abb. 2.2. Bei der Analyse der Klinikdaten wird sich aber im Kap. 7 herausstellen, dass die Datenanalyse vom Zeitpunkt der Durchführung abhängt.

Eine Zustandsanalyse ohne Berücksichtigung der Zeit mit adaptiven Verfahren auf den MEDAN-Daten findet man speziell im Abschnitt 5.4 (ohne Regelgenerierung) und ausführlicher in den Kapiteln 5 und 6 (mit Regelgenerierung) beschrieben.

Berücksichtigt wird die Zeit dann bei der Analyse im Kap. 7. Das Schema einer Zustandsanalyse unter Berücksichtigung der Veränderungen in der Zeitreihe zur Alarmgebung kann man sich anhand der Abb. 2.2 verdeutlichen. Zu den Zeitpunkten, an denen die Blitze zu sehen sind, ist eine Warnung für den Arzt besonders sinnvoll. In den Zeiträumen  $K_1$ ,  $W_2$  und  $K_2$  kann ebenfalls eine Warnung an den Arzt gegeben werden. Im Abschnitt 7.5 werden die Warnungen sogar noch zwischen „kritisch“ und „sehr kritisch“ unterschieden.

## 2.2 Neuronale Netze

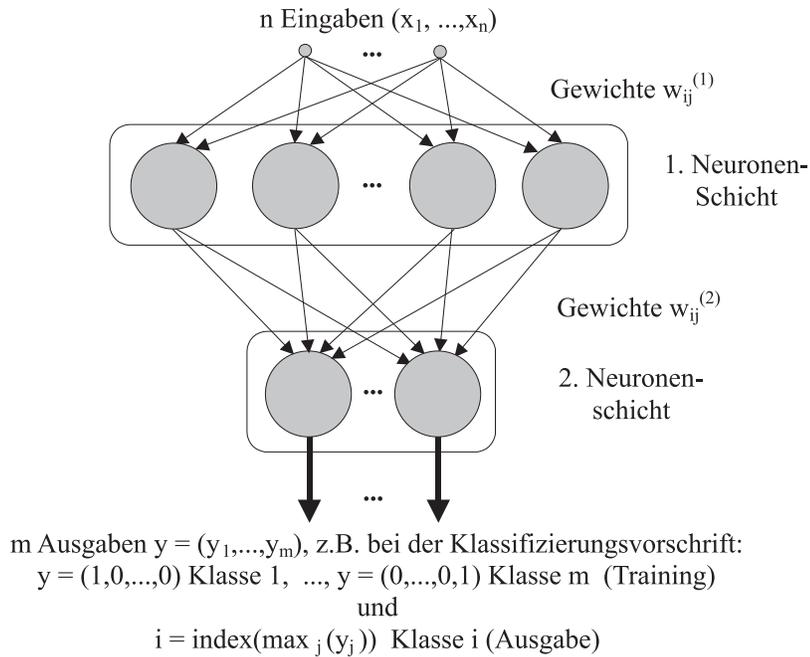
In diesem Abschnitt werden einige Paradigmen neuronaler Netze vorgestellt, die einer etwas genaueren Erklärung im Rahmen dieser Arbeit bedürfen. Neben den Grundbegriffen wird auf die wichtigsten Verfahren eingegangen. Interessant sind neuronale Netze als adaptive Architektur insbesondere zum Auffinden von Lösungen von Aufgabenstellungen auf kompliziert strukturierten (Realwelt-)Daten, die keine explizite statistische Modellierung erlauben. Aufgabenstellungen dieser Art sind z.B. die Klassifikation und die Regelerzeugung.

### 2.2.1 Grundlagen - Aufbau neuronaler Netze

Die Grundlagen neuronaler Netze (siehe [Bra95], [Hay99], [Roj93], [Zel94]) werden hier nur in aller Kürze dargestellt, um mit den verwendeten Ideen und Schreibweisen vertraut zu werden.

Ein neuronales Netz ist eine adaptive Architektur zur (annähernden) Lösung verschiedener Problemstellungen. Ursprünglich biologisch motiviert (Nachahmung der Funktionsweise von biologischen Neuronen im Gehirn), stehen mittlerweile auch die technischen, mathematisch-statistischen und anwendungsbezogenen Aspekte neuronaler Netze im Mittelpunkt der Forschung. Für die Datenanalyse ist nicht die Analogie zur Biologie entscheidend, sondern allein die Analyseleistung im Sinne einer mathematischen Abstraktion.

Neuronale Netze unterscheiden sich in ihrer *Netztopologie* und der Funktionsweise der einzelnen *Neuronen* (hier insbesondere *Lernregel* und *Aktivierungsfunktion*). Grundsätzlich unterschiedliche Lernverfahren sind *überwachtes* (engl.: supervised) Lernen, z.B. zur Approximation oder zur Klassifikation, und *unüberwachtes* (engl.: unsupervised) Lernen, z.B. zur Dichteschätzung oder zur Clusterung. Das *Reinforcement-Lernen* (verstärkendes Lernen) wird in [Hay99, Kap. 12] beschrieben, z.B. für Kontrollaufgaben. Die Netz-Architektur kann grob in vorwärtsgerichtete (engl.: feedforward, Abk.: FF) Netze und Netze mit Rückkopplungen (engl.: feedback, Abk.: FB) unterschieden werden. In der Abb. 2.3 ist exemplarisch ein FF-Netz mit seinen wichtigsten Komponenten dargestellt.



**Abbildung 2.3:** Architektur eines (überwachten) Feedforward-Netztes (zur Klassifikation).

Das neuronale Netz lernt (schematisiert) bei überwachten Lernverfahren iterativ in Zeitschritten, bis die Ausgabe zufriedenstellend ist, d.h. bis sich das Netz durch Anpassung der Netztopologie oder Parameter „adaptiert“ hat, also bis gilt:

$$\text{Fehler } \text{err} := \| \text{Ausgabe } y - \text{Sollausgabe } T \| < \text{Fehlerschranke } \varepsilon . \quad (2.1)$$

Einer der häufigsten vorkommenden Parameter bei neuronalen Netzen sind die sog. *Gewichte*, die angeben wie stark eine Verbindung zwischen Neuronen ist. Erreicht wird dann die Fehlerminimierung mit Hilfe einer fehlerbasierten Lernregel:

$$\text{Gewichte}_{\text{neu}} w_{ij}^{(k)}(t) := f(\text{Lernrate } \eta(t), \text{Gewichte}_{\text{alt}} w_{ij}^{(k)}(t-1), \text{Fehler } \text{err}(t-1)). \quad (2.2)$$

Selbstverständlich können in (2.2) zusätzliche Parameter berücksichtigt werden.

Da in dieser Arbeit überwachte Lernverfahren zur *Klassifikation* im Vordergrund stehen, sind die Ausführungen über das Training und Testen neuronaler Netze für diese Aufgabenstellung im Abschnitt 2.3.2 enthalten.

Bei unüberwachten Lernverfahren gibt es keinen Fehler, keine Sollausgabe und keine Fehlerschranke im o.g. Sinne. Hier wird solange gelernt bis das Lernverfahren

$$\text{Gewichte}_{\text{neu}} w_{ij}^{(k)}(t) := g(\text{Lernrate } \eta(t), \text{Gewichte}_{\text{alt}} w_{ij}^{(k)}(t-1)) \quad (2.3)$$

konvergiert, siehe z.B. das PCA-Lernverfahren im Abschnitt 2.4. – Die Verfahren können je nach Lernregel weitere (problemabhängige) Parameter besitzen, die das Lernverhalten entscheidend beeinflussen können und oftmals vom Experten voreingestellt werden müssen.

In den folgenden Abschnitten wird wichtige Literatur zu verschiedenen Verfahren neuronaler Netze bereitgestellt. Die in dieser Arbeit verwendeten Verfahren werden ausführlicher dargestellt, die anderen aufgrund des Vorhandenseins bekannter Literatur nur knapp. Die vielen Eigenschaften der neuronalen Netze und viele weitere Hinweise auf Spezialliteratur entnehme der interessierte Leser der angegebenen Literatur.

Ein grundlegendes Lernverfahren ist das *Gradientenverfahren* zur *Optimierung*, das auf der Tatsache basiert, dass eine (partiell differenzierbare) Funktion  $f : U \subset \mathbb{R}^n \rightarrow \mathbb{R}$ ,  $U$  offen, in Richtung des negativen Gradienten  $-\text{grad}_x f(x) = -\nabla_x f(x) = \left(\frac{\partial f}{\partial x_1}(x), \dots, \frac{\partial f}{\partial x_n}(x)\right)$  sich einem lokalen Minimum nähert, was insbesondere bei einer Fehlerfunktion von Bedeutung ist. Gradientenverfahren gibt es in allerlei Varianten, die die Konvergenz beschleunigen oder das Steckenbleiben in einem *lokalen* Minimum vermeiden. Einige Varianten werden in [Bis95, Kap. 7] beschrieben.

Im Zusammenhang mit neuronalen Netzen kann man das Gradientenverfahren als Lernregel (mit Zielfunktion  $f$  und Lernrate  $\gamma(t)$ ) wie folgt implementieren, siehe [Bra95, S. 69-72]:

$$\Delta w := w(t-1) - \gamma(t) \text{ grad}_w f(w(t-1)) . \quad (2.4)$$

### 2.2.2 Backpropagation

Die überwachten Lernverfahren *Gradientenabstieg* und *Backpropagation* für FF-Netze werden umfangreich in [Hay99, Kap. 4], [Bra95, S. 240-258], [Bis95, Abschnitt 4.8], [NKK96, Kap. 5] und [Zel94, Kap. 8 und 9] behandelt (nebst einiger Varianten zur Behebung der Probleme „lokale Minima“, „flache Plateaus“).

Die Architektur mehrschichtiger FF-Netze wird meist als  $x_1 - x_2 - \dots - x_n$  angegeben; „ $x_i$ “ bedeutet  $x_i$  Neuronen in der  $i$ -ten Schicht. Backpropagation ist ein überwachtetes Lernverfahren, das auf der Berechnung eines Fehlers beruht, die entgegengesetzt zur Richtung der Aktivierung durchgeführt wird. – Die starre Netz-Architektur kann durch Pruning (Entfernen von Gewichten und Neuronen, z.B. [Zel94, Kap. 24]) bzw. Einfügen von Neuronen (siehe z.B. [Leh99]) flexibilisiert werden.

Mehrschichtige FF-Netze mit der Lernregel Backpropagation (überwachtes Lernen) werden für viele Aufgaben eingesetzt: Funktionsapproximation, neuronale Faktorenanalyse, Klassifikation, Sensitivitätsanalyse, ICA.

[Sim95, S. 269-270] stellt zwei medizinische Anwendungen von FF-Netzen (trainiert mit Backpropagation) dar: Diagnose erblicher Muskelkrankheiten und Diagnose von Ohrenschmerzen. [BDHH97] beschreibt die Vorhersage von colorektalen Krebspatienten mit einem FF-Netz (ebenfalls trainiert mit Backpropagation). Viele weitere Anwendungen für spezielle medizinische Probleme existieren (siehe z.B. [Rap97]).

### 2.2.3 Radiale-Basis-Funktionen-Netze

Radiale-Basis-Funktionen-Netze (Abk.: RBF-Netze) eignen sich prinzipiell für die gleichen Lernaufgaben wie Backpropagation-Netze (allgemeiner Funktionsapproximator). Es handelt sich allerdings um ein *lokales Lernverfahren*. Wesentlich für die Funktionalität sind die gewählten (radialen) *Basisfunktionen*, z.B. Gaußsche Kerne, und die Wahl der *Zentren* (Platzierung der Mittelpunkte der Kernfunktionen). Literaturangaben zu RBF-Netzen (und deren Varianten) sind:<sup>4</sup> [Hay99, Kap. 5], [Bra95, S. 286-318], [Bis95, Kap. 5], [Zel94, Kap. 20], [NKK96, Kap. 6], [Fri98, Kap. 12 und 13], [HJ01a] und [HJ01b].

Es folgt eine kleine Auswahl von Anwendungen, die mit RBF-Netzen realisiert wurden:

- In [PB99] wird mit RBF-Netzen das Gummiprofil in der Reifenherstellung untersucht.
- In [Bla98, Kap. 5] werden RBF-Netze zur Patientenvorhersage im Vergleich zum SAPS II verwendet (beide mit Ergebnissen ähnlicher Güte).
- In [THV94] werden RBF-Netze zur Systemidentifikation nichtlinearer Systeme verwendet.

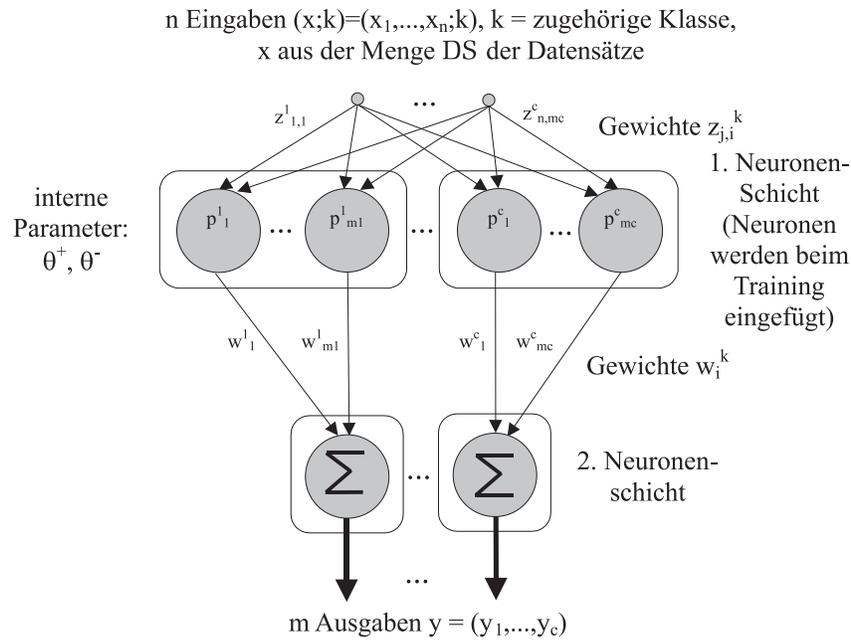
An dieser Stelle seien noch kurz einige Bestandteile eines RBF-Netzes genannt.  $x$  sei der Eingabevektor,  $z_i$ ,  $i \in \mathbb{N}$ , seien die Zentren,  $R_i = R_i(\|x - z_i\|)$  die Zentrumsfunktionen (z.B. *Glockenfunktionen* als Basisfunktionen; bekannteste Basisfunktion: Gaußsche Glockenfunktion) und  $W := (w_{ij})$  die Gewichte (vom  $i$ -ten Neuron der verdeckten Schicht zum  $j$ -ten Neuron der Ausgabeschicht).

Die Initialisierung, d.h. die Lage der Zentren, kann im Voraus durch einen Clusteralgorithmus geschehen oder in einen überwachten Mechanismus integriert werden. Während des Lernens können die Parameter der Basisfunktionen verändert werden (z.B. kleinere Radien), aber auch die Anzahl der Zentren kann variiert werden. Außerdem wird die Gewichtsmatrix  $W$  adaptiert.

Ein zweischichtiges RBF-Netz kann man als Spezialfall der sog. *Support Vector Machine* (Abk.: SVM) ansehen. Die SVM ist prinzipiell für die gleichen Problemstellun-

---

<sup>4</sup>Dort wird auf die zahlreichen Fachartikel der vergangenen Jahre verwiesen.



**Abbildung 2.4:** Architektur des RBF-DDA-Netzes.

gen wie ein RBF-Netz einsetzbar. Die SVM verwendet dabei Methoden der mathematischen Optimierungstheorie, siehe [Hay99, Kap. 6] oder [Sch97]. Vorteile scheint die SVM vor allem bei technischen Problemen bei hoher Datengüte zu besitzen, d.h. bei Problemen, bei denen selbst eine sehr kleine Performanzverbesserung wichtig ist.

Da im Kapitel 3 auf die Regelgenerierung (und die Klassifikation) mit *Fuzzy-RecBF-DDA-Netzen* sowie auf Verbesserungen davon eingegangen werden wird, erfolgt an dieser Stelle eine einheitliche Beschreibung der Verfahren *RBF-DDA* ([BD95], [Ber97, Kapitel 4] oder [BD98]), dass auch separat zur Klassifikation eingesetzt wird, *RecBF-DDA* [BH95] und später im Kapitel 3 dann *Fuzzy-RecBF-DDA* [HB95]. Diese Verfahren bauen von ihren Ideen her aufeinander auf und werden hier in einer einheitlichen Schreibweise vorgestellt.

Das RBF-DDA-Netz, das insbesondere für Klassifikationsaufgaben geeignet ist, verwendet in der ersten Schicht radiale Basisfunktionen, hat jedoch in beiden Schichten *separate* Neuronen für jede Klasse (Abb. 2.4). Es hat einen stärkeren *probabilistischen* Charakter, ähnlich zum *Probabilistischen Neuronalen Netz* (Abk.: PNN, [Spe90] oder [Zel94, Kap. 21]), vermeidet aber den Nachteil des hohen Speicheraufwandes des PNN, das ein Neuron pro Datenpunkt verwendet. Das Lernverfahren des RBF-DDA-Netzes baut auf der Idee der RCE-Netze [RCE82] (Abk. für: Restricted Coulomb Energy) auf.

**Das RBF-DDA-Netz:**

Parameter:

$R_i^k$  ist die Aktivierungsfunktion mit Radius  $\sigma_i^k$  des  $i$ -ten Prototyps  $p_i^k$  der  $k$ -ten Klasse, siehe (2.5) im Anschluss an den Algorithmus. Die Standardwahl der Schwellen  $\theta^+$  und  $\theta^-$  ist  $\theta^+ := 0.4$  und  $\theta^- := 0.2$ .

Lernphase einer Epoche:

1. Gewichte zurücksetzen:

```

for  $k = 1$  to  $c$ 
  for  $i = 1$  to  $m_k$ 
     $w_i^k := 0$ ;
  end
end

```

2. Muster durchgehen:

```

for  $r = 1$  to  $\text{size}(\mathcal{DS})$ 
  if  $R_i^k(x_r) \geq \theta^+$  für ein  $i \in 1, \dots, m_k$ ,
3. Muster durch  $p_i^k$  überdeckt (cover):
     $w_i^k := w_i^k + 1$ ;

```

4. Neues Neuron einfügen (commit):

```

else
   $m_k := m_k + 1$ ;
   $w_{m_k}^k := 1.0$ ;
   $z_{m_k}^k := x$ ;

```

5. Radius adaptieren:

$$\sigma_{m_k}^k := \min_{l \neq k, 1 \leq j \leq m_l} \left\{ \sqrt{-\frac{\|z_j^l - z_{m_k}^k\|^2}{\ln \theta^-}} \right\};$$

```

end

```

6. Konfliktneuronen verkleinern (shrink):

```

for  $l \neq k$ 
  for  $j = 1$  to  $m_l$ 
     $\sigma_j^l := \min \left\{ \sigma_j^l, \sqrt{-\frac{\|x - z_j^l\|^2}{\ln \theta^-}} \right\}$ ;

```

```

end

```

```

end

```

```

end

```

Im Schritt 3 verändert [BH95] willkürlich nur das Gewicht *eines* Neurons, auch wenn die Bedingung auf mehrere Neuronen zutreffen kann. Es besteht aber kein Unterschied bez. der Performanz des Netzes, falls man das Gewicht all dieser Neuronen

erhöht, so wie ich es später für das modifizierte Fuzzy-RecBF-DDA-Netz machen werde.

Ausführungsphase (symmetrische Radien):

$$1. \text{ Schicht: } R_i^k(x) := \exp\left(-\frac{\|x - z_i^k\|_2^2}{(\sigma_i^k)^2}\right), \quad (2.5)$$

$$2. \text{ Schicht: } y_k(x) := y_k(R_i^k(x)) := \sum_{i=1}^{m_k} w_i^k R_i^k(x) \quad (2.6)$$

und

$$\text{normalisierte Klassenwahrscheinlichkeit: } P(k|x) = \frac{y_k}{\theta^- + \sum_{l=1}^c y_l} \quad (2.7)$$

Interessant ist die Eigenschaft der Konvergenz des RBF-DDA-Lernverfahrens auf den Trainingsdaten, insofern man voraussetzt, dass keine identischen Trainingsvektoren mit unterschiedlicher Klassenzugehörigkeit existieren [Ber97, S. 37-38]. Diese Voraussetzung ist allerdings in der Praxis nicht unbedingt erfüllt, siehe hierzu den Abschnitt 3.1.

Die Idee mit Hilfe einer DDA-Architektur Regeln zu generieren, führt im nächsten Schritt zur Verwendung von *rechteckigen* Aktivierungsfunktionen.

### Das RecBF-DDA-Netz:

Parameter:

Für alle Dimensionen  $d$  benötigt man zusätzlich zu den RBF-DDA-Parametern einen Parameter  $\sigma_{d,\min}$ , der zu „schmale“ Regeln vermeiden hilft, d.h. der  $\sigma_i$ ,  $i = 1, \dots, n$ , nach unten begrenzt; er kann als Standard (bei normalisierten Daten in  $[0, 1]$ ) auf 0.1 gesetzt werden. -  $K^\infty$  sei die Menge derjenigen Radien,<sup>5</sup> die *unendlich* werden dürfen;  $K^f$  sei die Menge der endlichen Radien.<sup>6</sup> Im Laufe des Verfahrens können als unendlich definierte Radien von  $K^\infty$  in die Menge  $K^f$  durch die shrink-Operation wechseln.

Eine Erweiterung stellt die Verwendung zweier Radiensätze  $\sigma_{i,+}^k$  und  $\sigma_{i,-}^k$  mit  $\sigma_{i,+}^k$  und  $\sigma_{i,-}^k$  pro Prototyp  $p_i^k$  anstelle des einen Radiensatzes  $\sigma_i^k$  dar. Man erhält dann *asymmetrische* Rechtecke, die zu einer Verbesserung der Regelgenerierung führen, siehe [BH95].

---

<sup>5</sup>Strenggenommen haben Rechtecke keine *Radien*, sondern *Ausdehnungen* in die einzelnen Dimensionen; in Analogie zu den RBF-DDA Netzen wird aber weiterhin von Radien gesprochen.

<sup>6</sup>„f“ steht für *finite*.

Lernphase:

Der Lernalgorithmus unterscheidet sich im shrink-Mechanismus. Es wird diejenige Dimension zur Verkleinerung des Radius ausgewählt, die das größte Volumen des verbleibenden Rechtecks erhält. Dabei ist eine Fallunterscheidung zwischen den Radien aus  $K^\infty$  und  $K^f$  nötig (vgl. hierzu später die Bem. 3.1.3).

Ausführungsphase (symmetrische Radien):<sup>7</sup>

$$1. \text{ Schicht: } R_i^k(x) := \min_{1 \leq d \leq n} A(x_d, z_{d,i}^k, \sigma_{d,i}^k), \quad k = 1, \dots, c, \quad i = 1, \dots, m_k, \quad (2.8)$$

$$A(x_d, z_{d,i}^k, \sigma_{d,i}^k) := \begin{cases} 1 & , |x_d - z_{d,i}^k| \leq \sigma_{d,i}^k \\ 0 & , \text{sonst} \end{cases} \quad (2.9)$$

2. Schicht: gewichtete Summe (2.6) analog zum RBF-DDA-Netz

und Regelgenerierung:

$$i\text{-te Regel } \mathbf{if} \forall 1 \leq d \leq n : x_d \in (z_{d,i}^k - \sigma_{d,i}^k, z_{d,i}^k + \sigma_{d,i}^k) \mathbf{then} \text{ Klasse } k \quad (2.10)$$

Diese Form von Regeln gilt für endliche Gebiete; erlaubt man unendliche Radien, so kann man die für die Regel wichtigen Dimensionen selektieren. Eine unwichtige Dimension  $d$  ist eine mit der Regel: **if** ... **and**  $x_d \in (-\infty, \infty)$  **and** ... **then** Klasse  $c$ .

In [BF00] werden RBF-DDA- und RecBF-DDA-Netze zur Regelgenerierung eingesetzt, allerdings wird als Güte nur die Anzahl der Regeln, nicht aber die Klassifikationsleistung und die Abdeckung der Daten durch einzelne Regeln betrachtet, vgl. den Abschnitt 2.5.1. Dies wird ausführlich im Kap. 3 für das von mir verbesserte Fuzzy-RecBF-DDA-Netz nebst einer Beschreibung des Verfahrens nachgeholt. Im Abschnitt 2.3.2 wird noch auf einige Aspekte der Klassifikation und des Lernens von Klassifikationen eingegangen werden.

Ergänzung: Ein etwas anderer Ansatz als in [BH95], [HB95] (Fuzzy-Rec-BF-DDA) aus dem gleichen Zeitraum stammt von Tschichold-Gürman (*Fuzzy RuleNet*), beschrieben z.B. in [NKK96, Abschnitt 20.3] oder in [TG97]. Abweichend von [BH95] werden in [TG97] mit einer Delta-Lernregel *kontinuierliche* Gewichte gelernt. Es gibt keinen cover-Mechanismus. Der Überlappungsgrad der Regeln wird manuell eingestellt. Es gibt keine Regeln, die in einer Dimension unendliche Ausdehnungen zulassen. [TG97] beschreibt eher ein fehlerbasiertes Standard-Lernverfahren (verwandt mit dem Verfahren aus [Nau99], das auf einem Fuzzy-Perceptron beruht), [BH95] eher eine geometrische, wahrscheinlichkeitsorientierte Adaption.

## 2.2.4 Selbstorganisierende Karten und U-Matrix-Methode

*Selbstorganisierende Karten* (engl.: self-organizing maps, Abk.: SOM, auch: *Kohonen-Karten*) sind im Wesentlichen durch Kohonen, z.B. [Koh97], bekannt geworden.

<sup>7</sup> $d$  sei eine von  $n$  Dimensionen des Datenraums.

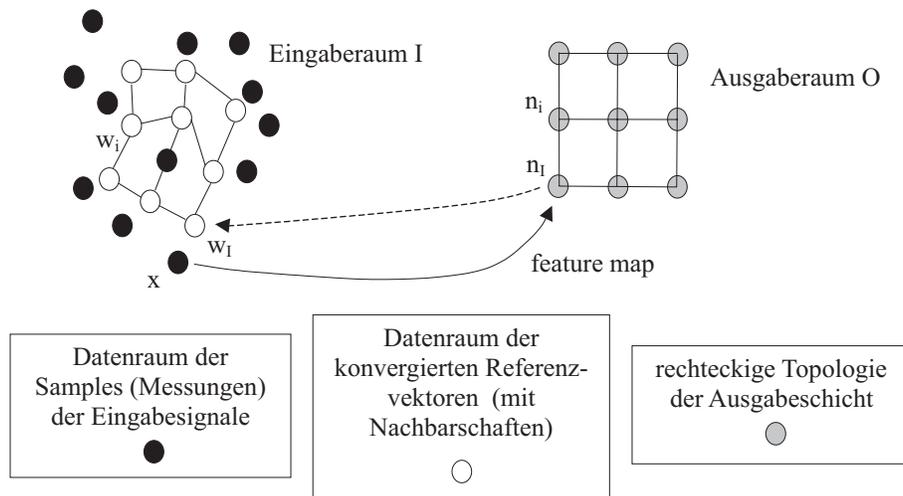


Abbildung 2.5: Das Schema einer SOM.

Es existieren zahlreiche Varianten dieses unüberwachten Lernverfahrens, das auf einer Nachbarschafts-erhaltenden Karte basiert, die sich im Datenraum entfaltet. Die Knoten der Karte repräsentieren nach Beendigung (Konvergenz) des Lernverfahrens den Datenraum. Einen Überblick über Kohonen-Netze geben auch [Hay99, Kap. 9], [Bra95, S. 140-169] und [Zel94, Kap. 15]. Verwandt mit SOM, aber ohne Nachbarschaften ist die Lernende Vektorquantisierung (Abk.: LVQ), siehe [Zel94, Kap. 14] oder [Koh97, Kap. 6].

Die SOM-Lernregel wird kurz wiedergegeben (nach [Koh97, S. 86-88]): Der Eingaberaum  $\mathcal{I}$  sei eine Teilmenge von  $\mathbb{R}^n$ . Jedem Knoten<sup>8</sup>  $i$  aus einem Ausgaberaum  $\mathcal{O}$  sei ein *Referenzvektor*  $w_i = (w_{i1}, \dots, w_{in})$  zugeordnet, siehe die Abb. 2.5. Die Knoten sind in einer Topologie, z.B. einer rechteckigen, angeordnet.  $x(t)$  sei der Eingabevektor zum Zeitpunkt  $t$ . Sei  $I$  der *Gewinner-Knoten*, d.h.  $I := \text{index}(\min_i \{\|x(t) - w_i(t)\|\})$ . Die *Nachbarschaftsfunktion*  $N(I, i; t)$  sei z.B. durch eine Gaußsche Funktion  $N(I, i; t) := \eta(t) \exp\left(-\frac{\|n_I - n_i\|^2}{2\sigma^2(t)}\right)$  gegeben ( $\eta$  Lernrate,  $n_I$  und  $n_i$  Ortsvektoren aus  $\mathcal{O}$  der Knoten  $I$  bzw.  $i$ , oft aus  $\mathbb{R}^2$  gewählt). Die Gewichte  $w_i(t)$  zum Zeitpunkt  $t = 0$  seien zufällig gewählt. Die Lernregel lautet dann:

$$w_i(t+1) := w_i(t) + N(I, i; t)(x(t) - w_i(t)) . \quad (2.11)$$

Einige Varianten und Anwendungen der SOM sind:

- Wachsende Kohonen-Netze (Growing Grid) werden in [Fri98, Abschnitt 9.3] vorgestellt.

<sup>8</sup>Die Bezeichnung *Knoten* wird hier synonym zu *Neuron* verwendet.

- In [Rit93] wird die *Parametrized Self-Organizing Map* eingeführt. Trotz weniger verwendeter Knoten soll mit Hilfe einer Interpolation eine möglichst stetige Ausgabe erreicht werden.
- [Bau96] beschreibt, wie der sog. *magnification factor* (dt.: Vergrößerungsfaktor)  $\mu$  kontrolliert werden kann. Damit erreicht man im Falle  $\mu = 1$  die Adaption der SOM-Knoten an die Dichte der zugrundeliegenden Daten, allerdings mit deutlich erhöhtem Laufzeitverhalten.
- In [Gui98], [GU99] wird ein SOM-Ansatz vorgestellt, der die (Cluster-)Struktur multivariater Zeitreihen medizinischer Daten erlernt.
- [VWG00] gibt einen aktuellen Überblick über einige Anwendungen der SOM in der Medizin, insbesondere zur Dimensionsreduktion.

Weitere Anwendungen werden in [Koh97, Kap. 7] beschrieben. – Eine theoretische Arbeit über Topologieerhaltung bei SOM, die auch frühere Arbeiten aufgreift, ist [Vil96].

Die *U-Matrix-Methode*<sup>9</sup> [US90] wurde als Visualisierung von Kohonen-Netzen eingeführt. Größere Abstände der Kohonen-Knoten führen zu größeren Matrixeinträgen, die als Clustergrenzen visualisiert werden können, je nach Höhe in verschiedenen Farben. Da sie auch für medizinische Anwendungen eingesetzt wurden, möchte ich sie an dieser Stelle einführen. Dabei werden die Schreibweisen beibehalten mit dem Zusatz, dass die Knoten  $n_i \in \mathbb{R}^2$  in einem Gitter im Ausgaberaum mit zugehörigen Gewichten  $w_{ij}$  angeordnet sind und  $d$  eine Metrik (Abstandsmaß) im Eingaberaum bezeichnet.

Geht man von einer ausgelernten zweidimensionalen  $(n \times n)$ -Kohonen-Karte  $\mathcal{K}$  mit quadratischem Gitter aus, so wird die *U-Matrix* wie folgt definiert:

### 2.2.1 Definition: (U-Matrix, [US90] oder [Ult91, Abschnitt 7.3])

a) Die U-Matrix wird definiert mit Hilfe folgender Abstände ( $i, j \in \mathbb{N}_n$ ):

$$d_x(i, j) := d(w_{ij}, w_{i+1,j}) , \quad (2.12)$$

$$d_y(i, j) := d(w_{ij}, w_{i,j+1}) , \quad (2.13)$$

$$d_{xy}(i, j) := d(w_{ij}, w_{i+1,j+1}) , \quad (2.14)$$

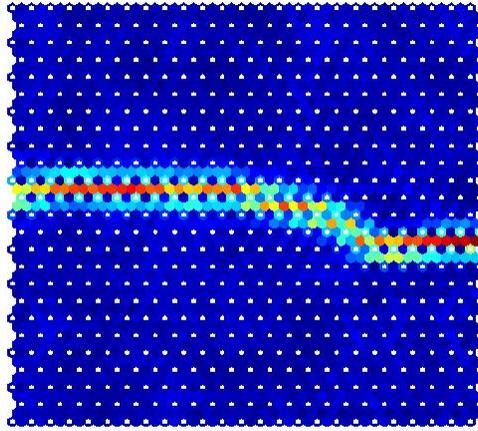
$$d_{yx}(i, j) := d(w_{i,j+1}, w_{i+1,j}) , \quad (2.15)$$

$$d_z(i, j) := \frac{1}{2}(d_{xy}(i, j) + d_{yx}(i, j)) \text{ und} \quad (2.16)$$

$$d_u(i, j) := 0 . \quad (2.17)$$

---

<sup>9</sup>Das „U“ steht für *unified distance*.



**Abbildung 2.6:** U-Matrix zweier getrennt liegender dreidimensionaler Normalverteilungen.

b) Dann definiert man die **U-Matrix**  $U \in \mathbb{R}^{(2n-1) \times (2n-1)}$ :

$$U := \begin{pmatrix} d_z(1,1) & \dots & d_y(1,j) & d_z(1,j) & \dots & d_y(1,n) & d_z(1,n) \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots \\ d_x(i,1) & \dots & d_u(i,j) & d_x(i,j) & \dots & d_u(i,n) & d_x(i,n) \\ d_z(i,1) & \dots & d_y(i,j) & d_z(i,j) & \dots & d_y(i,n) & d_z(i,n) \\ \vdots & \dots & \vdots & \vdots & \dots & \vdots & \vdots \\ d_x(n,1) & \dots & d_u(n,j) & d_x(n,j) & \dots & d_u(n,n) & d_x(n,n) \\ d_z(n,1) & \dots & d_y(n,j) & d_z(n,j) & \dots & d_y(n,n) & d_z(n,n) \end{pmatrix}. \quad (2.18)$$

Zu beachten ist, dass für eine  $(n \times n)$ -Karte eine  $((2n-1) \times (2n-1))$ -Matrix entsteht.

Anschließend werden die Eingabedaten durch  $\mathcal{K}$  abgebildet und zusammen mit dem Ergebnis der U-Matrix und dem Kohonen-Gitter visuell (V) dargestellt.

Dabei werden „Täler“ in (V) als Cluster angesehen, die durch „Mauern“ getrennt sind. Dabei treten einige Probleme auf, wie z.B. unterschiedliche relative Höhen oder unterbrochene Mauern, die eine Interpretation der Cluster erschweren (siehe [UH91, Abschnitt 4.3]). Die U-Matrix-Methode kann man auch im Sinne einer *Multidimensionalen Skalierung* (Abk.: MDS) sehen, da multivariate Daten auf eine zweidimensionale Karte abgebildet werden. Zwei Anwendungen sind:

- Clusterung von Blutmesswerten (siehe [Ult91, Abschnitt 4.4]) und
- Lawinenprognose (siehe [Sch95]).

Weitere Visualisierungsmethoden von SOM werden in [Ves99] beschrieben und veranschaulicht. In der Abb. 2.6 ist als Beispiel die U-Matrix zweier getrennt liegender dreidimensionaler Normalverteilungen wiedergegeben.<sup>10</sup> Im Sinne einer Clusteranalyse (vgl. Abschnitt 2.3.1) kann man die „vertikale Mauer“ als Clustergrenze interpretieren, so dass in der Abb. 2.6 zwei getrennt liegende Cluster auf der zweidimensionalen Ebene zu sehen sind.

Leider hat sich nach verschiedenen Versuchen die Hoffnung nicht erfüllt, auch bei den metrischen MEDAN-Daten Cluster mit der U-Matrix-Methode zu finden. Das Ergebnis ist allerdings nicht der Methode anzulasten. Auch hierarchische Clusteranalysen (vgl. Abschnitt 2.3.1) haben in [Pau01] keine sinnvollen Cluster auf den Daten gefunden. Die Erklärung hierfür liegt in *kontinuierlichen* Datenänderungen begründet. Es gibt keine *sprunghaften* Änderungen in den Daten, die zu Clustern führen.

Ergänzung: In den Arbeiten von Fritzsche, zusammengefasst in [Fri98], werden die *Growing Cell Structures* (Abk.: GCS) betrachtet. Es handelt sich um eine Erweiterung der Kohonen-Netze dahingehend, dass Knoten und Kanten eingefügt werden können, die Netzstruktur also variabel ist, aber die Netzdimensionalität fest bleibt. Dabei werden Triangulierungen des Datenraumes vorgenommen, die allerdings entarten können. Die GCS sind bez. ihrer Netztopologie *flexibler* als Kohonen-Karten durch das Einfügen von Knoten. [CZ00] präsentiert einen Zusatz für das Einfügen von Knoten *außerhalb* des triangulierten Knotennetzes mit einem Benchmark auf den medizinischen Proben-Daten.

### 2.2.5 (Wachsende) Neuronale Gase

Ähnlich wie ein Kohonen-Netz, nur ohne *feste* Netztopologie, funktioniert das *Neuronale Gas* (Abk.: NG) [Mar92] (oder [Bra95, S. 159-163]). Die Knoten breiten sich erst aus, anschließend können Nachbarschaften ausgebildet werden. Als Ergänzung zum NG werden im *Growing Neural Gas* (Abk.: GNG) Knoten eingefügt (und entfernt); dabei werden die Nachbarschaften während des Lernvorgangs gebildet. Der Lernvorgang von NG und GNG wird in [Fri98, Kap. 8] anschaulich verglichen. Eine überwacht arbeitende Variante des GNG ist das Supervised GNG (Abk.: SGNG, [Fri94], [Fri98, Abschnitt 13.2]), bei dem der Knoten-Kanten-Struktur RBF-Knoten aufgesetzt werden und der Ausgabefehler zum Trainieren des Netzes mit einer Delta-Lernregel benutzt wird. In [Bru98] werden *Dynamic Cell Structures* (Abk.: DCS) eingeführt und untersucht. Sie funktionieren ähnlich wie das (S)GNG. Die Ergebnisse einer Klassifikation ausgewählter Vorstudien-daten mit einem SGNG werden im Abschnitt 5.4 kurz dargestellt und mit denen einer Klassifikation mit RBF-DDA verglichen.

---

<sup>10</sup>Die Abb. 2.6 wurde mit dem Programm „SOM Toolbox Version 1.0 beta“, erstellt, vgl. den Anhang A.5.

Ergänzung: Obige Verfahren funktionieren mit *stationären* Daten (*stationäre Zellstrukturen*). Die Idee, bei *nicht-stationären* Datenmengen ein GNG einzusetzen, bei dem das Ausfügen von Knoten möglich ist, wird in [Fri97a] (GNG-U, Abk. für: GNG with *utility criterion*) oder auch in [Ham01] aufgegriffen. In diesem Zusammenhang werde ich von *nichtstationären* Zellstrukturen sprechen.<sup>11</sup> Da sich Patientenkollektive im Laufe der Jahre verändern können (veränderte Lebensbedingungen und Behandlungsmethoden) kann der Einsatz von nichtstationären Zellstrukturen in einer medizinischen Anwendung lohnen, die *dauerhaft* eingesetzt werden soll. Andererseits muss der Nachweis der dauerhaften Funktionsfähigkeit durch weitere, praxisrelevante Experimente bestätigt werden.

## 2.3 Zustandsanalysen

Drei wichtige grundlegende Methoden der Zustandsanalyse, die auch im Zusammenhang der in dieser Arbeit durchgeführten Analysen zu sehen sind, werden an dieser Stelle diskutiert: *Clusteranalyse*, *Klassifikation* und *Zeitreihenanalyse*. Im Rahmen dieser Arbeit steht eher die spezifische Anwendung der Verfahren der Zustandsanalyse auf den MEDAN-Daten im Vordergrund, nicht etwa der Vergleich der zahlreichen Verfahren untereinander.

### 2.3.1 Clusteranalyse

Jeder hat eine mehr oder weniger grobe Vorstellung davon, was ein *Cluster*  $C$  (dt.: Klumpen oder Häufung) innerhalb einer Datenmenge  $D$  ist. Man muss aber ein wenig aufpassen: Die Datenmenge  $D$  kann aus den verschiedensten Datentypen bestehen, z.B. (reellen) Messwerten, Booleschen Werten (Ja, Nein) oder auch nominalen Kenngrößen wie z.B. Berufe. Die Verrechnung dieser verschiedenen Arten von Daten ist aber mehr ein Problem der Skalierung. Eng verbunden mit dem Problem der Clustersuche ist das Problem der Dichteschätzung, da bei beiden Problemstellungen eine Datenmenge geeignet repräsentiert werden muss.

Prinzipiell beruht die Vorgehensweise einer Clusteranalyse im Auffinden von *ähnlichen* (homogenen) Untermengen  $C_i \subset D$ , wobei jede Menge  $C_i$  für alle  $i \neq j$  *unähnlich* (heterogen) von den Mengen  $C_j$  sein sollte. Die Ähnlichkeit innerhalb einer Menge  $C_i$  wird mit einem *Ähnlichkeitsmaß*  $\nu_1(C_i)$  gemessen; die Unähnlichkeit zwischen zwei Mengen  $C_i$  und  $C_j$  wird mit einem *Unähnlichkeitsmaß*  $\nu_2(C_i, C_j)$  gemessen. Mit Hilfe von Schwellwerten als Parameter für die Ähnlichkeit und Unähnlichkeit kann man festlegen, wann eine Teilmenge von  $D$  zu einem *Cluster* von  $D$  wird. Dabei kann man unterscheiden, ob die Durchschnitte der Mengen  $C_i$  leer sind oder nicht, d.h.  $\forall i \neq j : C_i \cap C_j = \emptyset$  oder  $\neq \emptyset$ . Im letzteren Fall kann man den Punkten innerhalb der Cluster Wahrscheinlichkeiten für eine gewisse Zugehörigkeit zu den Clustern vergeben.

---

<sup>11</sup>Zu diesem Thema existieren weitere Ansätze, z.B. die *ART-Netze*, vgl. [Zel94, Kap. 22].

Die verschiedenen Verfahren unterscheiden sich im Wesentlichen in der Wahl der Ähnlichkeits- und Unähnlichkeitsmaße.

Das Ergebnis einer Clusterung kann eine Nummerierung der Cluster sein, so dass den Punkten aus der Menge  $D$  die Nummer eines Clusters zugeordnet werden kann (und evtl. eine Wahrscheinlichkeit für die Zugehörigkeit zu einem Cluster). – Nach der Durchführung einer Clusterung müssen die Ergebnisse sorgfältig interpretiert werden, z.B. kann man zwei Clustern die Bedeutung „gesunder Mensch“ oder „kranke Mensch“ zuordnen, wenn man von Patientendaten ausgeht.

Es sollen nun kurz einige wichtige Verfahren der Clusteranalyse aus der Statistik beschrieben werden. Anschließend soll auf Möglichkeiten einer neuronalen Clusteranalyse eingegangen werden. Eine Einführung in statistische Clusteranalyseverfahren (single- oder complete linkage, k-means) geben [HE95, Kap. VII], [Bac96], [Boc73] oder [DH73, S. 213-248]. Zu beachten ist auf jeden Fall, dass verschiedene Clusterverfahren *verschiedene* Ergebnisse liefern, die einer Interpretation durch den Anwender bedürfen. Hierarchische Verfahren (linkage) können durch baumartige Strukturen, *Dendrogramme* genannt, dargestellt werden, siehe z.B. [DH73, S. 228-237].

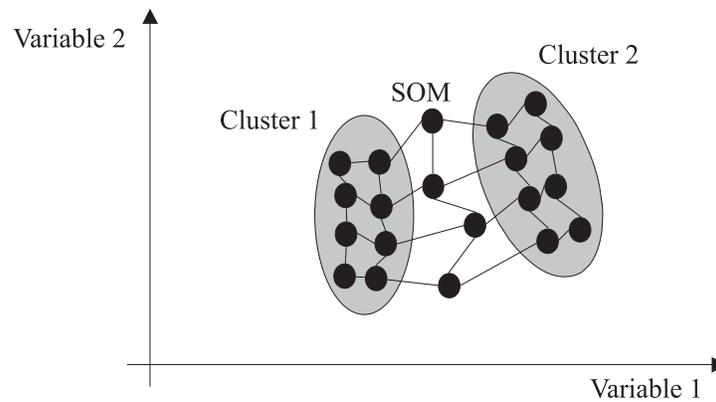
Ein etwas anderer Ansatz ist die Fuzzy-Clusteranalyse, [HKK97] oder [Zim91, Kap. 11]. Ein bekanntes Beispiel für eine Fuzzy-Clusteranalyse ist das Fuzzy- $c$ -means-Verfahren aus den 70-er Jahren. Hier werden Methoden aus dem Gebiet der Fuzzy-Mengen verwendet, um Überlappungen bei der Clusterbildung zuzulassen. – Ein Gitter-basiertes Clusterverfahren wird in [SE97] vorgestellt. Hier wird der Datenraum  $D$  geschickt mittels Gitter (variabler Länge) eingeteilt.

Für eine neuronale Clusteranalyse eignen sich prinzipiell unüberwachte neuronale Lernverfahren wie Vektorquantisierung und SOM (siehe Abb. 2.7 und Abschnitt 2.2.4), (wachsende) Zellstrukturen, (wachsende) neuronale Gase (siehe den Abschnitt 2.2.5) sowie Varianten davon. Die entsprechenden neuronalen Netze wurden im Abschnitt 2.2 vorgestellt.

Hat man eine Clusterung mit einer SOM durchgeführt, so ist es möglich die Clusterstruktur mit Hilfe der sog. *U-Matrix-Methode* aus Abschnitt 2.2.4 zu visualisieren, siehe [US90]. Anschließend kann man mit der so gewonnenen Clusterung Regeln extrahieren, siehe [Ult91], [UH91] oder in der medizinischen Anwendung [UKK95].

Als Anwendungsgebiete einer Clusteranalyse seien folgende exemplarisch genannt:

- allgemein: Marketing (Gruppierung von Konsumenten) oder Medizin (Patienten- oder Variablengruppierung) [HE95],



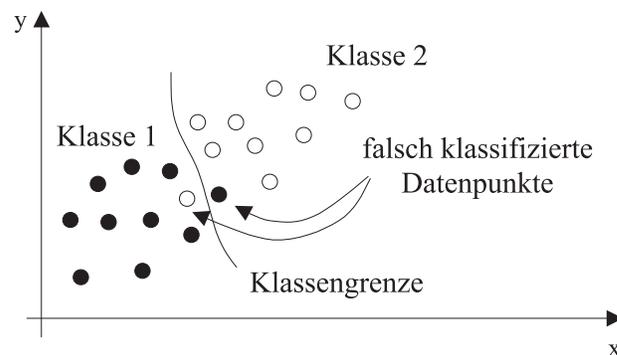
**Abbildung 2.7:** Schema einer Clustering mit einer SOM. Zwei Cluster werden entdeckt.

- Erkennung von japanischen Phonemen mit selbstorganisierenden Karten [Koh97, S. 241-242]
- Clustering von Dokumenten nach textuellem Inhalt [MT95],
- Clustering medizinischer Daten mit einer SOM [VWG00, S. 138-151] und
- Clustering von Proteinsequenzen gleicher Funktionalität in der Bioinformatik [GK01].

Eine linkage-Clustering auf den MEDAN-Vorstudiendaten wird in [HPT<sup>+</sup>00] beschrieben. Diese hat ebensowenig wie die Anwendung der U-Matrix-Methode medizinisch verwertbare Ergebnisse liefern können. Eine Clustering zum Auffinden von ähnlichen Regeln wird im Abschnitt 6.3 eingeführt.

### 2.3.2 Klassifikation

Neben der *unüberwachten* Clustering ist die *überwachte* Klassifikation ein wichtiges Datenanalyse-Instrument. Hat man zu einer vorgegebenen Klassifikationsinformation  $\mathcal{K} = \{class_1, \dots, class_n\}$  für eine Datenmenge  $D$  mit einem Klassifikationsverfahren eine Klassifikation berechnet, so dass alle Datenpunkte  $x \in D$  genau einer Klasse  $class_i$  angehören bzw. mehreren Klassen mit unterschiedlichen Wahrscheinlichkeiten, so können weitere Datenpunkte  $\tilde{x}$  aus einer anderen Menge  $\tilde{D}$ , z.B. aus einer Testmenge, ebenfalls den gefundenen Klassen zugeordnet werden. Für die Daten aus  $D$  oder  $\tilde{D}$  kann ein Fehler berechnet werden, der sog. *Klassifikationsfehler* (siehe die Abb. 2.8). Beispielsweise kann man anhand einer Menge von Patienten, die einer bestimmten „Krankheitsklasse“ angehören, Bereiche bestimmen, in denen



**Abbildung 2.8:** Schema einer Klassifizierung im zweidimensionalen Raum. Schwarz = Klasse 1, weiß = Klasse 2.

vorrangig Patienten einer Klasse vertreten sind. Anschließend kann man „neue“ Patienten klassifizieren.

Die Klassifikation kann man als ein Teilgebiet der *Mustererkennung* ([DH73] oder [The89]) ansehen, wie auch die Cluster- oder Zeitreihenanalyse. Entscheidend für ein gutes Klassifikationsergebnis ist die *Merkmalswahl* (engl.: feature selection), d.h. die Auswahl der Variablen (aus einem realen oder abstrakten Datenraum).

Wichtige, teilweise schon lang bekannte, statistische Klassifikationsverfahren (unterschiedlicher Qualität) sind: Bayes-Klassifikation (optimal bei a-priori-Annahmen über die Verteilung der Daten, siehe [The89, S. 29-35]), Fishers lineare Diskriminanzanalyse (siehe [Rao73, S. 482-485], [Hay99, S. 201-202]) und *k*-nearest-neighbour (dt.: *k*-nächster-Nachbar, siehe [The89, S. 129-131]).

Prinzipiell eignen sich zur neuronalen Klassifikation überwachte neuronale Netze, z.B. FF-Backpropagation-, RBF-Netze, Supervised GNG oder SVM. Wichtig ist in diesem Zusammenhang die *Generalisierungsfähigkeit* eines Klassifikationsverfahrens, d.h. die Fähigkeit des Verfahrens, auch unbekannte Daten korrekt zu klassifizieren. Daher sollte man unbedingt die zu klassifizierenden Daten in *Trainings-* und *Testdaten* unterteilen und in der Lernphase nur mit den Trainingsdaten arbeiten („Training eines neuronalen Netzes“).

Mit den Testdaten, die für das Netz unbekannt sind, wird die Klassifikationsgüte überprüft. Einen zusätzlichen *Validierungsdatensatz* (bei einer vorherigen Dreiteilung des Datensatzes), der nicht zwingend erforderlich ist, kann man anstelle des Testdatensatzes während des Trainings zur Überprüfung der Abbruchbedingung verwenden. Durch die Verwendung von Test- bzw. Validierungsdaten wird insbesondere das sog. *overfitting* (dt.: Überanpassung) vermieden, d.h. das „Auswendiglernen“

der Trainingsdaten mit Verlust der Generalisierungsleistung auf den Testdaten. Das Phänomen der Überanpassung tritt besonders bei Backpropagation auf, ist aber auch für andere Verfahren ein grundsätzliches Problem.

Theoretische Untersuchungen stehen für spezielle Anwendungen mit unbekanntem Verteilungen selten zur Verfügung, so dass man oft experimentell arbeiten muss, vgl. z.B. [Pre94], [HH98]. In [HKLO96] werden statistische und neuronale Klassifikatoren (FF-Backpropagation-Netze) verglichen; ein Vergleich von Verfahren bei der Klassifikation von Herzkrankheiten findet man in [DLK99]. Ein FF-Backpropagation-Netz erzielt dort bessere Ergebnisse als lineare statistische Verfahren. Neuronale Netze sind gut geeignet, um komplizierte, nichtlineare Klasseneinteilungen zu finden ([SRV96], [Pae02b]) und werden deshalb im MEDAN-Projekt für metrische Daten eingesetzt, siehe die Kapitel 5 und 7.

In [HBBE98] werden FF-Backpropagation- und RBF-Netze (aus dem Programm-Modul NeuralConnection von SPSS) verwendet, um eine Klassifikation von septischer Schock-Daten zu berechnen. Leider wurden hier die Ergebnisse *falsch* interpretiert. Es wurde nur das *beste* Ergebnis auf *einem* Testdatensatz angegeben nach einer *festen* Unterteilung in Trainings-, Validierungs- und Testdaten, so dass das erhaltene (gute) Ergebnis von über 90% ein „Zufallstreffer“ ist, der nach entsprechender zufälliger Partitionierung der Daten nicht bestätigt werden konnte. Zudem kann man sagen, dass viel zu wenige Daten (etwa 50 Datensätze) verwendet wurden, da das Ergebnis auf Trainings- und Validierungsdaten stark differiert. Die tatsächliche durchschnittliche Klassifikationsleistung auf den Daten betrug im Experiment lediglich 60%, was im Bereich des trivialen Klassifikators liegt mit Schwankungen im Bereich 30% bis 95%. Der zitierte Artikel kann somit *nicht* als wissenschaftlich fundiertes Ergebnis angesehen werden. Er beweist aber, dass Anwender bei komplexen Fragestellungen gut beraten sind mit Experten aus dem Bereich der Datenanalyse zusammenzuarbeiten. – Der falsche Gebrauch und der Missbrauch von neuronalen Netzen in der Medizin wird in [SVS97] untersucht. Viele weitere medizinische Anwendungen neuronaler Netze sind in [Rap97] wiedergegeben. Dort wird auch deren (un)sinnvolle Verwendung diskutiert.

Ich werde in allen Analysen verschiedene Trainings- und Testdaten verwenden und Versuchswiederholungen durchführen, so dass die Standardabweichung des Klassifikationsergebnisses berechnet werden kann (als Ersatz für ein Konfidenzintervall). Zudem werde ich keine Daten von Trainingspatienten zum Testen verwenden, d.h. ich verwende immer *disjunkte* Patientenmengen. Das ist sinnvoll, weil in medizinischen Anwendungen, so wie ich sie hier durchführe, das System bei unbekanntem Patienten und nicht nur bei unbekanntem Daten generalisieren soll. Verwendet man nicht-disjunkte Patientenmengen, so verschafft man sich einen unerlaubten Vorteil für die Klassifikationsleistung, siehe [HPT<sup>+</sup>00]. In [HPT<sup>+</sup>00] wird auch die Verwendbarkeit der Kreuzvalidierung (engl.: cross-validation) als Teststrategie diskutiert.

Wenn man *keine* Annahmen über die Beschaffenheit der Datensätze macht, d.h. alle (das sind unendlich viele) Datensätze betrachtet, so kann kein Klassifikationsverfahren einem anderen vorgezogen werden, siehe z.B. [DH73, neue Aufl. 2000]. Bekannt ist diese Aussage als „No Free Lunch Theorem“. In der Praxis ist es deshalb sinnvoll, verschiedene Klassifikatoren auf typischen Datensätzen innerhalb einer praktischen Anwendung zu testen; die empirischen Ergebnisse können für andere Anwendungen anders aussehen. Eine Möglichkeit die Klassifikationsleistung bei umfangreicher repräsentativer Datenmenge geringfügig zu verbessern bei gleichzeitiger Erhöhung der Modellkomplexität besteht in der Anwendung von *Resampling*-Techniken, z.B. Ensemble Averaging oder Boosting ([Hay99, Kap. 7], [DH73, Kap., neue Aufl. 2000]).

Ergänzung: Die Disziplin des *Algorithmischen Lernens* [Sch01] versucht allgemeine Aussagen über die Lernkapazität von Verfahren *unabhängig* von den Daten zu treffen. Ein sog. *PAC-Lernverfahren* (Abk. für: Probably Approximately Correct) ist ein Lernverfahren, das unabhängig von der Datenverteilung eine Problemklasse erlernen kann. Ein Begriff aus dieser Disziplin ist die sog. *VC-Dimension* (Abk. für: Vapnik, Chervonenkis), die die Erlernbarkeit der maximalen Anzahl beliebiger Trainingsdatenpunkte im Lernraum charakterisiert. Z.B. kann ein Kreis in  $\mathbb{R}^2$  immer alle Untermengen zweier verschiedener Punkte einschließen, drei Punkte aber nicht (z.B.  $(\dim_1, \dim_2; \text{Klasse}) = (0,0;1), (0,1;0), (0,2;1)$ ; dann lassen sich nicht nur der erste und dritte Punkt gemeinsam durch einen Kreis einschließen). Daher hat ein Kreis die VC-Dimension 2 im  $\mathbb{R}^2$ . Der Leser mache sich als kleine Übungsaufgabe klar, dass ein Rechteck im  $\mathbb{R}^2$  die VC-Dimension 4 hat (Lösung: [Sch01, S. 43-44]). Das Verfahren RBF-DDA hat die VC-Dimension  $2m$ , wenn  $m$  die Anzahl der verwendeten Neuronen ist [Ber97, S. 38-45].

Beispiele für Klassifikationsaufgaben sind:

- medizinische Diagnosen wie *Patient leidet an Krankheit A, Patient leidet an Krankheit B*, oder *großes Sterberisiko, geringes Sterberisiko*, siehe hierzu z.B. die Diagnose erblicher Muskelkrankheiten in [Sim95] oder die Entdeckung von Herzkrankheiten in [DLK99]. Weitere medizinische Beispiele werden in [Rap97] aufgeführt.
- Bilderkennung, z.B. Gesichtserkennung, oder Spracherkennung.<sup>12</sup>
- Psychologie, Soziologie (Klassifizieren von Menschen nach bestimmten Verhaltensmerkmalen).
- Klassifikation von Sprachstörungen durch neuronale Netze [AJBK00].

Die vorgegebenen Klassen sind bei den septischen-Schock-Patienten „verstorben“ und „überlebend“.<sup>13</sup> Herauszufinden, ob ein Patient stirbt oder überlebt, ist eine

<sup>12</sup>Zur Bild- bzw. Spracherkennung gehören viele Problemstellungen, die über eine *einfache* Klassifikation hinaus gehen, z.B. Kantensegmentierung oder Filterung von Störgeräuschen.

<sup>13</sup>Andere detailliertere Kategorien lagen von medizinischer Seite aus nicht vor.

schwierige Frage. Die Schwierigkeit liegt hier neben der Auswahl eines Klassifikators vor allem in der Datenvorverarbeitung und in der Auswahl von geeigneten Klassifikationsmerkmalen, siehe die Kapitel 5 bis 7.

Diese Fragen sind auch in der zeitlichen Entwicklung eines Patienten zu sehen, da seine Daten oft als Patientenkurve gegeben sind, so dass im nächsten Abschnitt in die *Zeitreihenanalyse* eingeführt werden soll.

### 2.3.3 Zeitreihenanalyse

Patientenakten enthalten Messungen für den Zeitraum des Krankenhausaufenthalts. Während einige Variablen nur einmal zu Beginn notiert werden (z.B. das Alter), werden andere Variablen wiederholt zu verschiedenen Zeitpunkten gemessen (z.B. Blutdruck). Insbesondere auf der Intensivstation werden für den Arzt wichtige Variablen regelmäßig gemessen (z.B. der Blutdruck alle 1 bis 2h). So entstehen *Zeitreihen*, deren Verlauf analysiert werden kann.

Bei einer Zeitreihenanalyse (z.B. [SS95]) geht es vorrangig um die *Prognose* von Werten und das *Erkennen* von typischen Verlaufsmustern.

Eine Prognose erstellen bedeutet aus  $n + 1$ ,  $n \in \mathbb{N}_0$ , aufeinanderfolgenden Werten  $x_t, \dots, x_{t+n}$  die weiteren  $m$ ,  $m \in \mathbb{N}_0$  Werte  $x_{t+n+1}, \dots, x_{t+n+m}$  vorherzusagen. Dabei ist i. Allg.  $m < n$ , siehe die Abb. 2.9. Ein typisches Verlaufsmuster z.B. von Zeitreihen mehrerer Patienten bez. einer gemessenen Variable tritt auf, wenn die statistischen (Werte Verteilung) und/oder die geometrischen Eigenschaften (Kurvenform) der Zeitreihen annähernd identisch sind.

Zeitreihen kann man als *stochastischen Prozess* interpretieren, d.h. eine Zeitreihe ist eine konkrete Realisierung eines stochastischen Prozesses. (Analog dazu ist ein reelles Zahlentupel eine konkrete Realisierung eines reellen Zufallsvektors.) Ein oft verwendetes Modell für Zeitreihen bzw. für stochastische Prozesse ist das *ARMA-Modell*, bei dem die früheren Werte Einfluss auf die aktuellen haben.

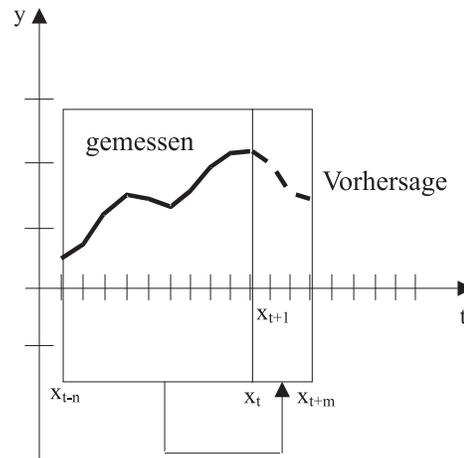
#### 2.3.1 Definition: (aus [SS95, S. 132])

Ein stochastischer Prozess  $(X_t)$  heißt **Autoregressiver-Moving-Average-Prozess** der Ordnung  $[p, q]$  (Abk.: ARMA $[p, q]$ ), wenn gilt:

$$X_t = \alpha_1 X_{t-1} + \dots + \alpha_p X_{t-p} + \varepsilon_t - \beta_1 \varepsilon_{t-1} - \dots - \beta_q \varepsilon_{t-q} . \quad (2.19)$$

Dabei ist  $(\varepsilon_t)$  ein White-Noise-Prozess [SS95, S. 95] und  $p, q \in \mathbb{N}$ .

Der *Box-Jenkins-Ansatz* zur Bestimmung der Modellparameter geht von einem ARMA-Modell aus und beinhaltet folgende Schritte (nach [SS95, S. 288]): Spezifikation der Parameter  $p$  und  $q$ , Grobschätzung, Feinschätzung, Diagnose, Interpre-



**Abbildung 2.9:** Schema einer Zeitreihenvorhersage mit einem neuronalen Netz. Mit 10 Werten werden 3 vorhergesagt.

tation, Anwendung des Modells. Speziell in der Medizin benötigt man *Vorverarbeitungsoperatoren* für *Monitor*-Daten, die prognostiziert werden sollen, wie Filterung, Interpolation, Differenzenbildung, siehe z.B. [Ste98, Kap. 3]. Die *nichtlineare* Zeitreihenanalyse ist in der Statistik ein Spezialgebiet; aufbauend auf den ARMA-Modellen wurden hierfür zahlreiche weitere Verfahren entwickelt, siehe [Ton90].

Prinzipiell eignen sich auch wieder alle überwachten Lernverfahren zum Erlernen einer Prognose, z.B. FF-Backpropagation, RBF (bez. NG siehe [MBS93]). Auf eine neuronale Zeitreihenanalyse wird z.B. in [Hay99, Kap. 13] oder [Bra95, Kap. 5] eingegangen. Als Eingabe eines neuronalen Prognose-Netzes sind die zeitlich aufeinanderfolgenden Werte  $x_{t-n}, \dots, x_t$  zu nehmen. Als Ausgabe erhält man die Prognosewerte  $\tilde{x}_{t+1}, \dots, \tilde{x}_{t+m}$ . Geachtet werden muss auf eine *echt vorhandene* Prognosefähigkeit eines neuronalen Netzes. Das Netz darf nicht die Zeitreihe auswendig lernen, so dass auch bei der Prognoseaufgabe mit Trainings- und Testdaten gearbeitet werden sollte.

Wichtige Anwendungen von Zeitreihenanalysen sind Prognosesysteme und Systeme, die bei einem kritischen Ereignis einen Alarm auslösen:

- Finanzdatenprognose (z.B. Vorhersage von Börsenkursen) [HH97],
- Erkennung von Atemstörungen [TLPD98]. Das Alarmsystem reguliert im Störfall die Beatmung,
- Erkennung von Ischämie (Minderdurchblutung) des Herzens mittels einer eindimensionalen SOM [FPB99],
- Ausreißererkennung beim Herzschlag und Anstieg des Blutdrucks [BGI98],

- U-Matrix-Methode zur Mustererkennung für Zeitreihen, angewendet auf medizinische Daten [GU99],
- diskrete Mustererkennung in Zeitreihen [AS95] auf der Basis von Assoziationsregeln und
- Auffinden von ähnlichen Mustern in Zeitreihen durch transformationsinvariante Ähnlichkeitsmaße, die anhand auffälliger Punkte in den Zeitreihen (sog. „Landmarks“) berechnet werden [PWZP00].

Bei den MEDAN-Daten ist die explizite Berücksichtigung der Zeitreihenstruktur für eine Klassifikation der *einzelnen* Messwerte nicht möglich, da nur die Klassen „verstorben“ und „überlebt“ pro Patient vorhanden sind. Es existieren keine objektiven Kriterien für einzelne Messwerte, die zur Klassifikation herangezogen werden könnten. Es bleibt nur die Möglichkeit alle Messwerte eines Patienten mit der gleichen Klasse zu versehen. Findet man so Klassifikationsbereiche (Zustände) mit einer eindeutigen Zuordnung zu einer Klasse, kann man zu den entsprechenden Zeitpunkten der Patientenzeitreihen einen Alarm generieren. Im Kap. 5 wird sich zeigen, dass das nicht so gut möglich ist. Im Abschnitt 7.2 werden unterschiedliche Klassifikationsergebnisse innerhalb von *Zeitfenstern* erzielt. Eine explizite *Prognose* der Zeitreihen oder gar eine Erkennung typischer *mehrdimensionaler* Verlaufsmuster erscheint z.Z. sehr schwierig quantitativ analysierbar zu sein aufgrund der sehr unterschiedlichen Liegedauern, des individuellen Verhaltens der Patienten und der unregelmäßigen Messungen (fehlende Werte). Einfache Kreuzkorrelationen oder Spektralanalysen helfen hier nicht weiter.

## 2.4 Abhängigkeitsanalysen

Ergänzend zu den Zustandsanalysen und der noch folgenden Regelgenerierung wird grundsätzlich auch eine Vorverarbeitung (medizinischer) Daten, z.B. mit Korrelationsanalysen, Hauptkomponentenanalysen oder Unabhängigkeitsanalysen in Betracht gezogen. Daher werden die Grundlagen und Ideen dieser Verfahren vorgestellt. Das Ziel einer Abhängigkeitsanalyse kann die Reduktion der Eingabedimension für ein anschließendes Verfahren (z.B. Klassifikation) sein, um den *Fluch der Dimension* (engl.: curse of dimensionality) für dieses Verfahren zu vermeiden, siehe [Bis95, S. 7-9]. Auch kann eine Transformation der Eingabedaten angestrebt werden, in der Hoffnung im transformierten Raum die Datenanalyse besser durchführen zu können. Im Falle einer anschließenden Regelgenerierung sind Transformationen allerdings problematisch, da dann ja die Regeln im transformierten, nicht direkt interpretierbaren Datenraum, entstehen; somit ist eine echte Dimensionsreduktion für eine Regelgenerierung wünschenswert.

Zunächst wird der Begriff des *Signals* als spezielle Zeitreihe für die Zwecke dieser Arbeit eingeführt, da einige Analysen, z.B. die ICA diesen Begriff voraussetzen. Da bei medizinischen Analysen Signale unterschiedlicher Art auftreten, soll auch der Begriff des *Signals* grundlegend differenziert werden.

### 2.4.1 Signale

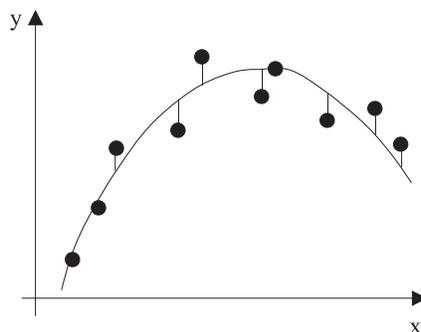
Es werden zunächst drei für uns wichtige Definitionen für ein *Signal* gegeben.

#### 2.4.1 Definition: (Signal)

Ein Signal ist ein Träger einer Nachricht. Beispiele hierfür sind Audio- und Videosignale, bekannt aus der Sprach- bzw. Bildverarbeitung. In dieser Definition soll angenommen werden, dass das Signal mit einer *angemessenen* Sampling-Rate vorliegt. Das Signal nenne ich unter diesen Umständen ein **Signal im engeren Sinn**. Für unsere Anwendungen liegen die Signale immer *zeitdiskret* vor. Unter einem **Signal im weiteren Sinn (Daten)** verstehe ich eine Datenmenge, von der man annimmt, dass sie keine oder keine angemessene Sampling-Rate besitzt und von der a priori nicht unbedingt klar ist, ob die gemessenen Werte tatsächlich Träger einer Nachricht sind. Elektrische, mechanische oder chemische Signale im engeren Sinn von lebenden Organismen nennt man **Biosignale**.

In [See97, Kap. 5] oder [vBM97, 8 und 25] werden (medizinisch bedeutende) Biosignale und ihre Analyse vorgestellt. Ein wichtiges Problem ist das der Diskretisierung der naturgemäß stetigen Signale. Das Abtast-Theorem von Shannon (siehe z.B. [Unb97]) gibt hierauf eine Antwort, die man folgendermaßen (ohne Formalismus der Dirac-Distribution) beschreiben kann: Die Abtastrate muss mindestens doppelt so hoch sein wie der höchste Frequenzanteil im Signal; z.B. soll die Abtastrate beim EEG mind. 60 Hz betragen, da im EEG-Signal relevante Frequenzanteile bis 30 Hz vorhanden sind. Bekannte Beispiele für Biosignale sind das Elektrokardiogramm (Abk.: EKG) und das eben genannte Elektroenzephalogramm (Abk.: EEG). Das EKG wird oft direkt neben dem Intensivpatienten an einem Monitor abgelesen (*online-monitoring*). Die Messung von Biosignalen ist insbesondere interessant, um eine grundsätzliche Modellvorstellung von der Funktionsweise lebender Systeme zu bekommen.

Die Bedeutung der Signalanalyse für das MEDAN-Projekt, z.B. als Mittel zur Datenvorverarbeitung, erscheint eher gering, obwohl die Biosignalanalyse (z.B. EKG- und EEG-Analyse) in der Medizin eine große Rolle spielt. Bei den uns zur Verfügung stehenden Daten des septischen Schocks handelt es sich um meistens täglich eingetragene Variablendaten, bei denen oft Werte fehlen. Von Biosignalen kann hier also keine Rede sein. Die MEDAN-Daten sind eher Signale im weiteren Sinne, also allgemeine *Daten*. Eine Messung von Monitor-Biosignalen innerhalb des MEDAN-Projekts ist auf medizinischer Seite aufgrund des Aufwandes nicht durchführbar.



**Abbildung 2.10:** Schema einer nichtlinearen Regression mit eingezeichneter Regressionskurve.

### 2.4.2 Statistische Standardverfahren

Alle im Folgenden genannten Verfahren (Korrelation, Regression, Hauptkomponenten- und Faktorenanalyse) sind in statistischer Standardsoftware enthalten und können auf allgemeine Daten angewendet werden. Sie werden nur knapp eingeführt unter Berücksichtigung medizinischer Beispiele und neuronaler Varianten, und die Verwendbarkeit für diese Arbeit wird geklärt.

**Korrelation:** Korrelationsanalysen sind in der Statistik wohlbekannt. Eindimensionale Analysen (Vergleich jeder Variable mit jeder anderen) sind besonders in der Medizin sehr beliebt, siehe [Har98] (zur mehrdimensionalen Korrelationsanalyse siehe [HE95, Kap. III]). Beispiele für Korrelationsanalysen werden in [Har93], [HE95], [Sac92], [Dap98] und [Har98] gegeben. Eine *lokale* Korrelationsanalyse (zwischen Wetterlage und Stromverbrauch), die die herkömmliche Korrelationsanalyse mit Kohonen-Karten verbindet, findet der Leser in [PG98]. Korrelationsberechnungen und darauf aufbauende Analysen auf den MEDAN-Daten findet der Leser vor allem in den Abschnitten 5.3 und 7.1.

**Regression:** Eine weitere wichtige Auswertungsmethode für (medizinische) Daten ist die Regression, siehe hierzu [Har93, S. 569-608], [HE95, S. 77-142] oder [Sac92, S. 489-577]. Prinzipiell wird bei einer Regression ein *funktionaler* Zusammenhang zwischen Variablen gesucht. Der einfachste Zusammenhang ist ein *linearer*.

Vermutet man keinen linearen Zusammenhang zwischen Variablen, so kann man versuchen, eine (statistische) polynomiale Regression als nichtlinearen Ansatz durchzuführen. Gesucht ist dann ein Regressionspolynom  $a_0 + a_1x + a_2x^2 + \dots + a_lx^l$  für  $l \in \mathbb{N}$ , das den Regressionsfehler minimiert. Der Fall  $l = 2$  wird in [Har93, S. 589-595], behandelt. Das Prinzip ist in Abb. 2.10 dargestellt.

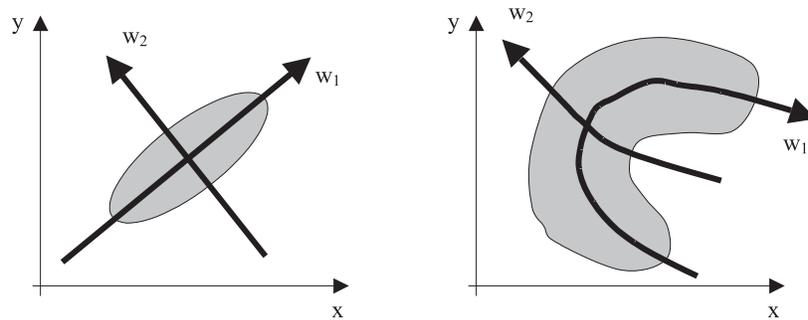
Es gibt zahlreiche weitere Varianten von nichtlinearen Regressionsverfahren, z.B. die *logistische* Regression, die sich besonders für binäre Ausgaben („krank“, „gesund“) eignet und in der Medizin eingesetzt wird. In [SRV96], [Pae02b] wird die logistische Regression mit der neuronalen Regression verglichen, wobei neuronale Netze eine etwas bessere Güte erreichen, wenn eine ausreichende Samplezahl vorliegt. In [Roj93, S. 186-188] und [Hay99, S. 84-89] werden die Zusammenhänge zwischen statistischer Regression und *Regressionsnetzen* besprochen. Ein neuronales Netz (z.B. ein zweischichtiges FF-Netz mit Backpropagation-Lernregel) kann so trainiert werden, dass es eine nichtlineare Regression durchführt, siehe z.B. [Bis95, S. 9-14]. Als Fehlermaß wird gewöhnlich der mittlere quadratische Fehler (engl.: mean squared error, Abk.: MSE) verwendet. Dabei ist wiederum auf eine zufällige Auswahl von Trainings- und Testdaten zu achten, um das overfitting zu vermeiden.

Auch selbstorganisierende Karten kann man als Regressionsnetze auffassen. Die Knoten einer selbstorganisierenden Karte platzieren sich nach einer gewissen Anzahl von Iterationsschritten innerhalb der vorgegebenen Datenwolke, wobei einige Neuronen nicht adäquat die vorgegebenen Daten repräsentieren (Regressionsfehler), vgl. das „Kaktus-Bild“ aus [Koh97, S. 94].

In der Standardliteratur findet der Leser zahlreiche Beispiele für Regressionsanalysen. Ein einfaches medizinisches Beispiel hierzu ist die vorhandene Beziehung zwischen Tabakkonsum und Lungenkrebs, das in [Har98] beschrieben wird. In [Dap98, S. 156-160] wird z.B. eine statistische und neuronale Regressionsanalyse für Gewässerdaten durchgeführt im Sinne einer *Sensitivitätsanalyse* zur Merkmalsauswahl. In [BBMM98] werden FF-Netze verwendet, die die Funktionalität einer logistischen Regression übernehmen, um eine (medizinische) Überlebenszeitanalyse durchzuführen. Für das MEDAN-Projekt bietet die statistische logistische Regression zur Klassifikation keine prinzipiellen Vorteile gegenüber einer Klassifikation mit neuronalen Netzen, so dass die metrischen MEDAN-Daten nur durch neuronale Verfahren analysiert werden.

**Hauptkomponentenanalyse:** Ein langbekanntes Verfahren zur Dimensionsreduktion (Pearson, 1901, sowie Hotelling, 1933) ist die *Hauptkomponentenanalyse* (engl.: Principal Component Analysis, Abk.: PCA). Die Theorie der (linearen) Hauptkomponentenanalyse ist im Standardwerk [Jol86] enthalten. Eine Übersicht über weitere Dimensionsreduktionsmethoden gibt [CPn96].

Da das Verfahren der *linearen* PCA selbst allgemein bekannt ist, wird hier nur kurz die Idee dargestellt. Gesucht ist eine Koordinatentransformation (in Form einer Matrix  $A$ ), die ein vorgegebenes Koordinatensystem  $\mathcal{K}_1$ , i.d.R. das Cartesische, in ein orthogonales Koordinatensystem  $\mathcal{K}_2$  transformiert. Eine Achse  $w_1$  von  $\mathcal{K}_2$  soll dabei in die Richtung der stärksten Varianz einer vorgegebenen Datenmenge  $D$  zeigen;  $w_1$  ist die sog. *1. Hauptkomponente*. Die weiteren Achsen stehen jeweils orthogonal



**Abbildung 2.11:** Schema einer linearen und nichtlinearen PCA mit eingezeichneten Hauptachsen bzw. Hauptkurven. Graue Fläche: Datenbereich.

aufeinander und zeigen unter dieser Nebenbedingung in die Richtung der absteigend schwächeren Varianzen der Datenmenge  $D$ . In [Jol86], aber auch in [Bra95], in [Hag97, S. 47-52], [Ger99, S. 136-138] oder [Hay99, 396-404], wird das Verfahren ausführlich erklärt. Es basiert auf der Berechnung der Eigenwerte der empirischen Korrelationsmatrix der Datenmenge  $D$ .

Ergänzung: Die *nichtlineare Hauptkomponentenanalyse* gibt die Vorstellung von Geraden als Hauptkomponenten auf. Ein Ansatz ist der von Hastie und Stuetzle [HS89], die sog. *principal curves* berechnen. Der nichtlineare Ansatz wird auch in [RMS91, S. 261-265] oder [CM98, S. 182-199] beschrieben. Dort werden Kohonen-Karten als Approximationsmethode für die Hauptkurven-Berechnung eingesetzt. In der Abb. 2.11 wird das Schema der linearen und der nichtlinearen PCA gegenübergestellt.

In der Arbeit [Hag97, Kap. 4], [Bra95] werden neuronale Algorithmen zur linearen PCA vorgestellt. Ein Beispiel für ein PCA-Netz (ohne laterale Verbindungen) ist das Netz von Sanger [San89]. Sangers Lernregel basiert auf der Hebbischen Lernregel und wird von ihm als *GHA* (Abk. für: Generalized Hebbian Algorithm) bezeichnet.

Die PCA wird generell zur Dekorrelierung und Variablenreduktion eingesetzt, siehe [Jol86] oder [Bra95], aber auch zur Bildkomprimierung, siehe [Bra95, S. 263-265] und [AB98]. In der Bildverarbeitung – denkbar ist z.B. die Verarbeitung von Röntgenbildern – fasst man hierzu die Bildpixel von Teilbildern zu Bildvektoren zusammen und führt hierauf eine PCA durch. Die PCA dient zur Vorverarbeitung einiger ICA-Algorithmen, die nur dekorrelierte Daten verarbeiten können, siehe dazu später den Abschnitt 2.4. Die PCA kann als Bestandteil einer Faktorenanalyse (siehe unten) verwendet werden. Falls der zu verarbeitende Datensatz vollständig vorliegt, kann man eine statistische PCA der neuronalen aufgrund der Berechnungsgeschwindigkeit und der Genauigkeit vorziehen. – Auf den Vorstudien Daten hat eine durchgeführte lineare PCA keine brauchbaren Ergebnisse geliefert, die zu einer Dimensionsreduzierung hätten führen können.

**Faktorenanalyse:** Die statistische Faktorenanalyse ist ähnlich wie die PCA ein langbekanntes Verfahren, dass seit einer Publikation von Spearman im Jahre 1904, zitiert in [Übe71], weiterentwickelt wurde. Bei der Faktorenanalyse geht es nicht allein um Reduktion von Variablen, sondern vielmehr um die Zusammenfassung von Variablen zu sog. *Faktoren*, z.B. könnte man Variablen Körpergröße und Gewicht zu einem Faktor „Statur“ zusammenfassen. Dabei ist die Variable „Statur“ selbst nicht messbar; solch eine Variable heißt *latent*. Die Faktorenanalyse ist insbesondere in der Psychologie ein oft verwendetes Werkzeug. Die historische Entwicklung der Faktorenanalyse wird in [Übe71] geschildert. Einen praktischen Einstieg in die Faktorenanalyse gibt [HE95].

Ergänzung: Es werden noch zwei weitere aktuelle Ansätze aus der Literatur genannt:

- Hauptfaktorenanalyse mit einem neuronalen Netz [Cha98] mit Anwendung auf einen Datensatz mit den Merkmalen Mord, Vergewaltigung, Raub, Überfall, Betrug, Diebstahl und Autodiebstahl. Es wird versucht, gemeinsame Faktoren zu finden.
- Independent Factor Analysis, basierend auf dem EM-Algorithmus (siehe [Att98]).

Eine Faktorenanalyse wurde nicht auf den MEDAN-Daten durchgeführt, da einerseits das Zusammenfassen von Merkmalen zu potenziellen Faktoren nicht das vorrangige Ziel ist und andererseits die durchgeführten Korrelationsanalysen einige Hinweise auf „zusammengehörige“ Variablen liefern.

### 2.4.3 Lineare ICA

Bei der Independent Component Analysis (Abk.: ICA, dt.: Analyse unabhängiger Komponenten) handelt es sich grob gesprochen um ein Verfahren, dass Signalmischungen entmischt, wobei die Signale ursprünglich aus unterschiedlichen voneinander möglichst stochastisch unabhängigen Quellen stammen sollen. Übersichten zu dieser Problematik geben z.B. [HK01], [Hay99, Kap. 10] und [Lee98]. Obwohl die ICA eher der Signalanalyse zuzuordnen ist, wurde der Versuch unternommen, die ICA als Datenvorverarbeitung (Dimensionsreduktion) auf die medizinischen MEDAN-Daten anzuwenden. Die Bewertung der Ergebnisse diesbez. wird am Ende des Abschnitts gegeben.

Da die ICA in der Medizinischen Informatik noch nicht weit verbreitet ist, wird sie zum besseren Verständnis illustriert (nach [Hay99, S. 523-525]), siehe die Abbildungen 2.12 bis 2.15.

Bislang wurde fast ausschließlich der lineare Fall (Abk.: LICA) im Sinne der folgenden Definition behandelt.

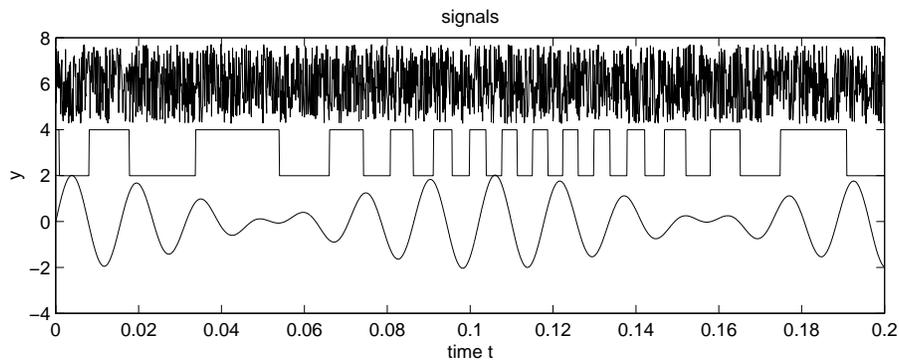


Abbildung 2.12: Drei ungemischte Signale.

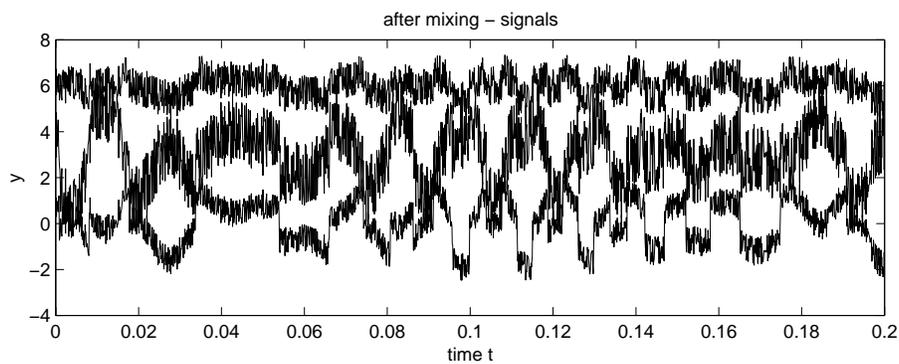


Abbildung 2.13: Drei mit einer linearen Abbildung gemischte Signale.

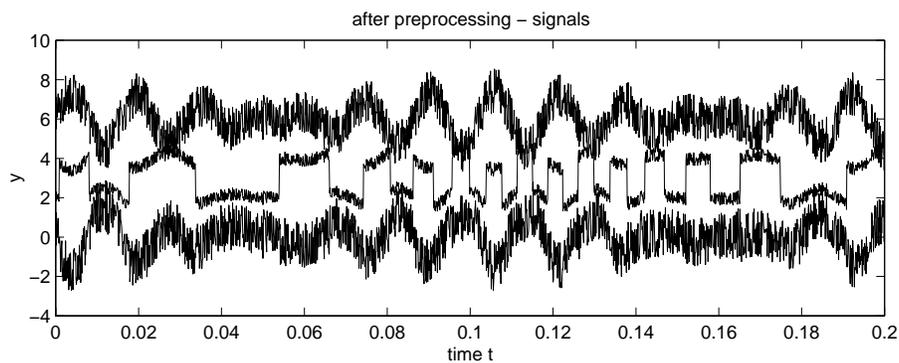


Abbildung 2.14: Drei Signale nach einer PCA.

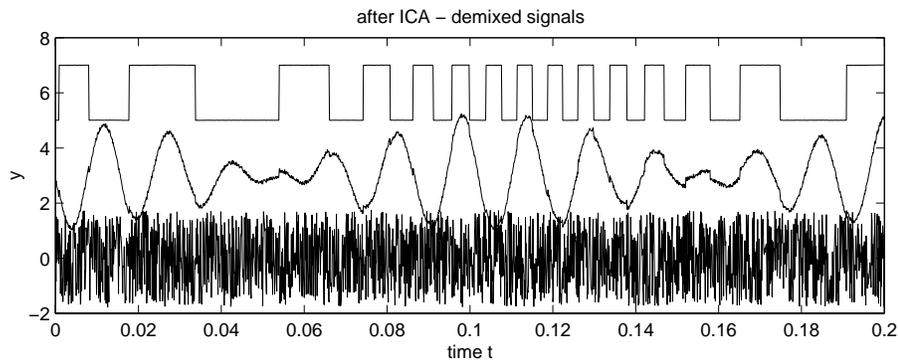


Abbildung 2.15: Drei Signale nach einer LICA.

### 2.4.2 Definition: (LICA)

Sei  $X$  ein Zufallsvektor und  $X = x$  eine Realisierung des Zufallsvektors  $X$ , d.h. konkret  $x \in \mathbb{R}^n$  für ein fest gewähltes  $n \in \mathbb{N}$ . Gesucht ist eine lineare Transformation, d.h. eine Matrix  $B \in \mathbb{R}^{m \times n}$  mit  $y := Bx$ , so dass die Komponenten  $y_i, i = 1, \dots, m$ , so stochastisch unabhängig wie möglich sind.<sup>14</sup>

Ergänzung: Allgemein nimmt man an, dass die Zufallsvektoren mittelwertfrei sind, d.h.  $E(X) = 0$ ; ansonsten zentriere man vorher gemäß  $X_{\text{neu}} := X_{\text{alt}} - E(X_{\text{alt}})$ . Für eine Realisierung  $X = x$  des Zufallsvektors  $X$  bedeutet das  $\langle x \rangle = 0$  bzw.  $x_{\text{neu}} := x_{\text{alt}} - \langle x_{\text{alt}} \rangle$ .

Die Definition 2.4.2 ist sinnvoll, da man annimmt, der Vektor  $x$  sei aus einer linearen Mischung  $x = As$  hervorgegangen und die Komponenten  $s_i$  seien stochastisch unabhängig. Dann bezeichnet man  $B$  auch als **Entmischungsmatrix** (engl.: demixing matrix). Ist dann  $B = A^{-1}$ , dann gilt für die Komponenten  $y_i = s_i$ , und sie sind stochastisch unabhängig. Diese Idealsituation erreicht man in der Praxis nicht. Störend wirken sich Datenfehler (z.B. Signalrauschen) aus bzw. Abweichungen von der Linearität der Mischung. Hinzu kommt eine den LICA-Verfahren inhärente Nicht-Eindeutigkeit der Lösung, die im folgenden Satz dargestellt wird:

### 2.4.3 Satz: (Comon, 1994)

a) Ist höchstens eine der Quellen des Zufallsvektors  $X$  normalverteilt, so ist das LICA-Problem lösbar.

b) Eine Lösung des LICA-Problems, d.h. eine Entmischungsmatrix  $B$ , ist eindeutig bestimmt bis auf

1. *Vorzeichen*, d.h. das entmischte Signal kann sich gegenüber dem Quellsignal im Vorzeichen unterscheiden,

<sup>14</sup>Zur Definition der stochastischen Unabhängigkeit siehe [Bau91, S. 53-54, 263].

2. *Skalierung*, d.h. bis auf Multiplikation mit einer invertierbaren Diagonalmatrix  $S$  mit reell positiven Einträgen und
3. *Reihenfolge*, d.h. bis auf Multiplikation mit einer Permutationsmatrix  $P$ .

Beweis: siehe [Com94], „theorem 11“ und „property 2“.

□

Die ICA ist der PCA als Datenvorverarbeitungsmethode insofern überlegen, da bei der PCA nur Momente 2. Ordnung<sup>15</sup> verwendet werden und aus der Dekorreliertheit nicht die stochastische Unabhängigkeit folgt,<sup>16</sup> vergleiche die Abb. 2.14 mit der Abb. 2.15.

Ergänzung: In [TLSH91] wird bewiesen, dass das Problem der Signaltrennung bis auf o.g. Nicht-Eindeutigkeiten auch gelöst werden kann, wenn man anstelle der Annahme der stochastischen Unabhängigkeit der Quellsignale  $s_i$  nur die *Unkorreliertheit* fordert. Allerdings muss man dann zusätzliche Annahmen über Zeitreiheneigenschaften der Signale berücksichtigen.

*Blind Source Separation* stellt die häufigste (elektrotechnische) Anwendung der ICA dar. Man geht hier davon aus, dass man *nichts* über die Art der Mischung der Quellen  $s_i$  weiß und somit *blind* eine Entmischungsabbildung finden muss, die allein auf den statistischen Eigenschaften der beobachteten Signale bzw. Variablen  $x_i$  beruht. I.d.R. betrachtet man genauso viele oder weniger Quellen wie Mischungen. Ein bislang allgemein ungelöstes, schwieriges Beispiel ist das sog. *Cocktail Party Problem*. Es handelt sich dabei um die Identifizierung mehrerer Sprecher bzw. Geräuschquellen in einem Raum. Das Problem wird in [Lee98, S. 97] illustriert. Das BSS-Problem kann man den sog. *inversen Problemen* unterordnen, bei denen die *direkte* Messung der Eigenschaften eines Objektes nicht möglich ist, sondern bei denen von Beobachtungen *indirekt* auf die interessierenden Größen zurückgeschlossen werden muss, um diese zu identifizieren.

Ein erfolgreicher Vertreter für LICA-Algorithmen, der grundsätzlich auf der Minimierung der Transinformation (vgl. [CT91]) zwischen den zu entmischenden Signalen beruht, ist der Fixpunktalgorithmus „FastICA“ [Hyv99]. Prinzipiell wird bei einer LICA eine Maximierung der Transinformation zwischen den Eingangs- und den Ausgangskanälen angestrebt, was äquivalent zur Minimierung der Transinformation aller Ausgangskanäle ist, allerdings werden dazu unterschiedliche (numerische) Methoden verwendet (zur Äquivalenz siehe [Lee98, Kap. 3]).

Bisherige Anwendungen der ICA umfassen hauptsächlich Probleme der Signalentmischung und BSS, z.B. die Stimmentrennung (bei Piloten im Cockpit oder bei

<sup>15</sup>Momente werden in [Hay99, Kap. 10], [KS69] definiert.

<sup>16</sup>Umgekehrt folgt aus der stochastischen Unabhängigkeit die Unkorreliertheit von Zufallsvariablen, siehe [Bau91, S. 53-54, 263].

Telefonkonferenzen), die Bildcodierung [AB98], die medizinische EEG- und EKG-Analyse [J<sup>+</sup>00] und die Zeitreihenanalyse (z.B. von Börsenkursen [BW97]).

Die LICA hat als Vorverarbeitung in [PMK96] bei einer Zeitreihenanalyse eines chaotischen Systems zu einer kleinen Verbesserung der Modellierung der Zeitreihen beigetragen. Eine Hypothese ist, dass man mit Hilfe der LICA als Vorverarbeitung zur Dimensionsreduktion bzw. Datentransformation auf den MEDAN-Daten einen Vorteil im Hinblick auf eine Klassifikation erreichen könnte. Dies hat sich allerdings nicht bestätigt, siehe den Abschnitt 3.2.3. Auch weitere Versuche blieben erfolglos. Eine Erklärung mag sein, dass die Annahme der Existenz stochastisch unabhängiger (Signal-)quellen, die vermischt wurden, i. Allg. bei Datensätzen nicht zutrifft.

Ergänzung: Der *nichtlineare Fall* ist wesentlich schwieriger zu handhaben, da nichtlineare Mischungen bedeutend komplizierter sein können als lineare und aufgrund der zahlreichen Möglichkeiten die Nicht-Eindeutigkeit der Lösung problematischer wird. Bei den wenigen bisher erschienenen Arbeiten über nichtlineare ICA (z.B. [YAC98]) werden i.d.R. einschränkende *Modellannahmen* getroffen, um erste Resultate zu erzielen. Diese Ansätze, die noch wenig erforscht sind, erfordern eine hohe Datenqualität und eine hohe Anzahl von (Signal-)Daten und funktionieren noch nicht bei einer größeren Variablenzahl, so dass hier von einer Anwendung abgesehen wird.

#### 2.4.4 Modellierung

Zum Abschluss dieses Abschnitts wird der Begriff des *Modells* eingeführt. Der Abschnitt bezieht sich auf alle in dieser Arbeit vorgestellten Verfahren, wenngleich er hier unter der Abhängigkeitsanalyse eingeordnet ist. – Verwendet man ein Verfahren zur Datenanalyse, trifft man explizit oder implizit eine oder mehrere *Modellannahmen*, z.B. wählt man ein Fuzzy-Modell mit bestimmten Zugehörigkeitsfunktionen, ein logistisches Regressionsmodell, ein lineares ICA-Modell oder ein Differentialgleichungsmodell 2. Ordnung. Die Auswahl eines geeigneten Verfahrens kann sehr schwierig sein: „In any data-driven modeling effort the two central tasks are always choosing the functional form of the model, and using the data to determine the adjustable parameters of the model“ [Ger99, S. 113]. Aufbauend auf dieser Aussage definiere ich kurz den Begriff des Modells:

**2.4.4 Definition:** Ein **Modell**  $\mathcal{M}(q_1, \dots, q_m)$  mit den Parametern  $q_1, \dots, q_m$  für eine Problemstellung  $\mathcal{P}(p_1, \dots, p_n)$  mit den Parametern  $p_1, \dots, p_n$  ist eine (mathematische) Abstraktion der Problemstellung  $\mathcal{P}$ . Die Lösung  $L \in \mathcal{L}(q_1, \dots, q_m)$  des Problems muss innerhalb der Modellannahme  $\mathcal{M}$  (algorithmisch) gefunden werden.

Grundlegende Eigenschaften von Modellen sind:

- quantitatives oder qualitatives Modell,
- diskretes oder stetiges Modell,

- stochastisches oder deterministisches Modell,
- dynamisches oder statisches Modell,
- lokales oder globales Modell,
- parametrisches oder nichtparametrisches Modell,
- Modelle mit und ohne a-priori-Annahmen.

Ein grundlegendes Prinzip zur Modellauswahl ist *Occams razor* (z.B. in [Bis95, S. 14-15]). Es besagt, dass man ein möglichst einfaches Modell wählen sollte, da ein einfaches Modell eine klarere Struktur besitzt. Z.B. möchte man bei der Regelgenerierung möglichst wenige Regeln zur Erklärung von Sachverhalten erzeugen (Redundanzvermeidung). Zwei Kriterien zur Auswahl eines einfachen Modells sind *MDL* (Abk. für: Minimum Description Length) oder *AIC* (Abk. für: Akaikes Information Criterion). Sie haben die Form:

$$\text{Kriterium zur Modellauswahl} := \text{Fehlermaß} + \text{Komplexitätsmaß} . \quad (2.20)$$

Unter diesen Gesichtspunkten wurden sinngemäß die Verfahren für die Zustandsklassifikation und Regelgenerierung für das weitere Vorgehen der Analyse des septischen Schocks ausgesucht. Bei einem fertig trainierten Fuzzy-RecBF-DDA-Netz (siehe den Abschnitt 3.1) handelt sich um ein quantitatives, stetiges, stochastisches, statisches, lokales, nichtparametrisches (=verteilungsunabhängiges) Modell. A-priori-Annahmen sind möglich in Form von vorgegebenen Regeln. – Ein erzeugtes Generalisierungsregelmodell (vgl. das Kapitel 4) stellt ein quantitatives, diskretes, stochastisches, statisches, lokales, nicht-parametrisches Modell dar, bei dem ebenfalls Regeln vorgegeben werden können.

Die Modellauswahl von verschiedenen Fuzzy-RecBF-DDA-Modellen bzw. Generalisierungsregelmengen, die jeweils auf dem gleichen Datensatz erzeugt wurden, wird im Abschnitt 3.7 bzw. 4.9 beschrieben.

Ergänzung: Hier werden noch Modelle genannt, die bisher nicht beschrieben wurden, interessant sind, aber in dieser Arbeit nicht verwendet werden.

- *Abhängigkeitsgraphen* oder *Bayes-Netze*: zur Darstellung von (probabilistischen) Abhängigkeiten von Daten-Entitäten [Whi90], [BH99, Kap. 4], z.B. Abhängigkeiten von Parametern bei Krebspatienten [BH99, S. 84].
- *Markov-Modelle*: zur Darstellung von Zustandsübergängen ([Hay99, S. 548-556] oder [Pap91, S. 635-654]); *Hidden-Markov-Modelle*: zur adaptiven Bestimmung eines Markov-Modells (auf den MEDAN-Daten: [Pau01]).
- Differentialgleichungen und Differentialgleichungssysteme (I): zur Identifikation eines dynamischen Vorgangs, z.B. Einfluss einer medikamentösen Dauertherapie einer Schwangeren auf den Fötus [Heu95, S. 490-491], Differentialgleichungssysteme für Krebszellenwachstum, Insulinsteuerung und Hämodialyse [Kra97].

- Differentialgleichungen und Differentialgleichungssysteme (II): QSIM (Abk. für: Qualitative Simulation, [Kui86]) zur Bestimmung des qualitativen Systemverhaltens dynamischer Systeme unter Berücksichtigung aller Nebenbedingungen; GENMODEL [HC93] zur Bestimmung derjenigen Nebenbedingungen, die ein vorgegebenes Systemverhalten ermöglichen.

## 2.5 Regelgenerierung

Die *Regelgenerierung* aus Daten dient dem *Verständnis* der Struktur eines Datensatzes, z.B. ergänzend zu einer Klassifikation wie sie im Abschnitt 2.3.2 beschrieben wurde. Der Anwender hat ein berechtigtes Interesse an den Gründen einer Klassifikationsentscheidung, z.B. kann ein Arzt nur aufgrund dieser Gründe eine neue Therapie entwerfen, nicht allein anhand einer Klassifikationsentscheidung. *Regeln* helfen dem Anwender bei der Reflexion der Ergebnisse. Die Regeln können dabei in einer der jeweiligen Anwendergruppe angepassten sprachlichen Form ausgegeben werden, wobei die grundsätzliche Form gegeben ist als: „**if** Eingabebedingung **then** Ausgabebedingung“. Wir betrachten nur Regeln der Form „**if** Bedingung(Variable<sub>1</sub>) **and** Bedingung(Variable<sub>2</sub>) **and** ... **and** Bedingung(Variable<sub>n-1</sub>) **and** Bedingung(Variable<sub>n</sub>) **then** Klassenzugehörigkeit“.

Um die anschließenden Kapitel vorzubereiten, werden im Folgenden die Grundlagen aus den Gebieten

1. Neuro-Fuzzy-Regelgewinnung sowie
2. Assoziations- und Generalisierungsregeln

vorge stellt und es wird auf Unterschiede der Verfahren hingewiesen. Dabei werden die Assoziations- und Generalisierungsregeln in einem einheitlichen Sprachgebrauch dargestellt. Abschließend wird dann im Abschnitt 2.5.3 auf eine Kombination der Verfahren eingegangen.

Ergänzung: Regelerzeugungsverfahren wie Entscheidungsbäume, evolutionäre Algorithmen, induktive logische Programmierung oder weitere logikbasierte Verfahren werden zwar am Ende des Abschnitts 2.5.2 kurz angesprochen, sie werden aber im Rahmen dieser Arbeit nicht weiter untersucht. Logikbasierte Verfahren sind zum Aufbau von Expertensystemen geeignet, bei denen Wissen vorgegeben wird, z.B. von Ärzten, weniger zum Lernen aus Daten. Evolutionäre Algorithmen sind geeignet, heuristisch suboptimale Lösungen zu einem vorgegebenen Ziel zu finden (z.B. Parameteroptimierung für möglichst niedrigen Klassifikationsfehler); diese Strategien können prinzipiell mit der Regelgenerierung kombiniert werden, vgl. S. 61. Entscheidungsbäume werden im Folgenden nicht betrachtet (mit Ausnahme des Abschnitts 4.4), da für die Klassifikation metrischer Werte neuronale Netze zur Verfügung stehen, deren Klassifikationsleistung über der von Entscheidungsbäumen liegt,

siehe z.B. die Vergleiche in [Ber97]. Die Idee der Assoziations- bzw. Generalisierungsregeln ist den Entscheidungsbäumen bez. symbolischer Variablen insofern überlegen, da mehrere Items *gleichzeitig* betrachtet werden, während im Entscheidungsbaum eine Variable nach der anderen betrachtet wird, und die Variablen in einer Hierarchie angeordnet sind. Das führt in hochdimensionalen, kompliziert strukturierten Räumen durch die Verwendung von *Thresholds* zu sehr unübersichtlichen Regeln, vgl. [BH99, Kap. 6]. Eine Kombination von symbolischen und metrischen Variablen ist im Entscheidungsbaum prinzipiell möglich, siehe dazu aber die Diskussion im Abschnitt 2.5.3.

### 2.5.1 Neuro-Fuzzy-Regelgenerierung

Neuronale Netze zeigen ein *black-box-Verhalten*, d.h. die innere Struktur bleibt dem Anwender verborgen. Es wird z.B. bei einer Klassifikation nicht klar, *warum* ein neuronales Netz eine bestimmte Klassifikationsentscheidung getroffen hat.

Insbesondere im Zusammenhang mit RBF-Netzen wurden sog. *hybride* Ansätze („Neuro-Fuzzy-Systeme“) entwickelt. Man lernt mit einem neuronalen Netz eine Klassifikationsentscheidung<sup>17</sup> und *defuzzifiziert* die Netzausgabe, d.h. man berechnet anhand der Netzausgabe eine für den Anwender verständliche Textausgabe aus einer endlichen Menge vorgegebener Textbausteine. In [JS93] wird die Äquivalenz der Aktivitäten der RBF-Neuronen mit *Fuzzy-Zugehörigkeitsfunktionen* (bestimmter Bauart) gezeigt. Damit ist ein Grundstein gelegt, um mit Hilfe neuronaler Netze Regeln zu generieren. Dem Lernvorgang kann man eine *Fuzzifizierung* vorschalten, um auch vorhandenes Wissen in das Netz zu integrieren.

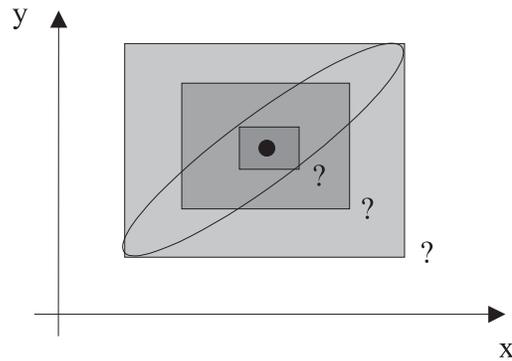
In [Ber97, S. 83-90] werden neben radialen Basisfunktionen auch *elliptische* Basisfunktionen untersucht; das Fazit ist allerdings, dass der geringe Vorteil bei der Regelgenerierung (etwas weniger Regeln) durch einen hohen Verwaltungs- und Zeitaufwand bei der Adaption der Ellipsen „erkauft“ wird (ein bis zu 100-facher Zeitaufwand). Außerdem steht man vor dem Problem, aus einer langgezogenen Ellipse mit einem Winkel ähnlich dem einer Winkelhalbierenden eine Regel zu generieren (Dilemma zwischen Nicht-Klassifikation und Fehlklassifikation), siehe Abb. 2.16. Man könnte zwar achsenparallele Ellipsen wählen, um diesem Problem zu begegnen, aber es bietet sich an, direkt mit Rechtecken oder Fuzzy-Rechtecken zu arbeiten.

Eine *überwachte* Regelgenerierung mit Klasseninformationen ist einer *unüberwachten* Regelgenerierung vorzuziehen, falls man die Klasseninformationen zur Verfügung hat, da Dichteschwankungen i. Allg. nicht die Klassengrenzen repräsentieren, siehe Abb. 2.17.

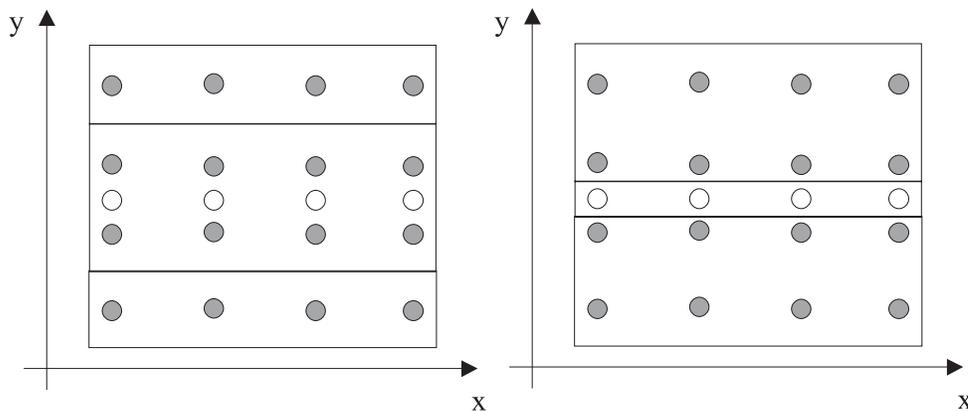
Symbolische Variablen, z.B. medizinische Begriffe, kann man mittels *vagen* Definitionen (z.B. „hoher Blutdruck“ anstelle „Blutdruck ist 185“) als *Fuzzy-Menge* modellieren. Mit einem *Fuzzy-System* oder *Neuro-Fuzzy-System* kann man Schlüsse

---

<sup>17</sup>Als wichtigste Anwendung betrachten wir im Folgenden immer die Klassifikation.



**Abbildung 2.16:** Problem der Regelgenerierung aus einer Ellipse: Welche rechteckige Regel repräsentiert am besten die Ellipse?



**Abbildung 2.17:** Regelerzeugung durch Clustering (links) und durch eine überwachte Strategie (rechts). Die ausgefüllten und nicht ausgefüllten Kreise sollen Datenpunkte zweier verschiedener Klassen repräsentieren. Die unüberwacht erzeugte mittlere durch Clustering erzeugte Regel beinhaltet Datenpunkte beider Klassen.

ziehen, z.B. medizinische Diagnosen erstellen. Eine Einführung in Fuzzy-Systeme gibt [KGGK93]; für Neuro-Fuzzy-Systeme konsultiere man [NKK96] oder [Nau00b]. Grundbegriffe und Beispiele findet man in [Bra95, Kap. 6], [Zim91] oder [BH99, Kap. 8]. Folgende Ansätze kann man unterscheiden; die Übergänge sind aber fließend:

- Eine Klassifikation mit anschließender Regelgenerierung (Regelgenerierung beeinflusst nicht die Klassifikation), z.B. Rechteckerzeugung aus trainierten nicht-rechteckigen RBF-Neuronen,
- Der Algorithmus ist auf Regelgewinnung ausgelegt und achtet nicht primär auf das Klassifikationsergebnis, z.B. Lernen mit rechteckigen anstatt mit radialen Basisfunktionen-Neuronen,
- Der Algorithmus versucht eine gute Klassifikationsleistung mit einer „guten“ Regelgenerierung zu verbinden wie z.B. die Fuzzy-RecBF-DDA-Methode.

Überschätzen sollte man aber nicht die Leistungsfähigkeit eines Neuro-Fuzzy-Systems, die oft an *beispielhaften* Anwendungen illustriert wird. Bei den Realweltdaten des septischen Schocks wird z.B. auf eine (De-)Fuzzifizierung der metrischen Variablen verzichtet, weil nur eine recht willkürliche Einteilung der Variablen in „hoch“, „normal“ oder „niedrig“ möglich gewesen wäre, die aber zu ungenau ist, um sinnvolle multivariate Regeln auf den Daten zu beschreiben. Aber auch ohne die (De-)Fuzzifizierungsmechanismen können interessante Regeln gelernt werden, siehe den Abschnitt 3.4.

Oft wird in der Literatur die Fuzzy-Begriffswelt mit der medizinischen Datenanalyse in Verbindung gebracht, so dass ich die wichtigsten Begriffe kurz erläutern möchte; auch werde ich den Begriff der Zugehörigkeitsfunktion im Kapitel 3 gebrauchen.

### 2.5.1 Definition: (Fuzzy-Mengen)

a)  $X, Y, Z$  bezeichnen im Folgenden (scharfe) Mengen und  $A, B, C$  *unscharfe*.  $A$  heißt dabei eine **unscharfe Menge** oder **Fuzzy-Menge** (engl.: fuzzy set), wenn eine **Zugehörigkeitsfunktion** (engl.: membership function)  $\mu_A : X \rightarrow [0, 1]$  existiert, die überall auf  $X$  definiert ist.  $\mu_A(x)$  heißt **Zugehörigkeitsgrad** (oder **Fuzzy-Niveau**) von  $x \in X$  zur Menge  $A$ . Beispiele für Zugehörigkeitsfunktionen sind Dreiecks- und Trapez-Funktionen. Eine Fuzzy-Menge kann man folgendermaßen notieren:  $A = \{x, \mu_A(x)\}_{x \in X} = \sum_{x \in X} \mu_A(x) / x$ . In der Abb. 2.18 ist eine zweidimensionale trapezförmige Zugehörigkeitsfunktion zu sehen. Ihr höherdimensionales Analogon kommt beim Fuzzy-RecBF-DDA-Verfahren zur Anwendung, siehe Kap. 3.

b) Standard-Operationen für Vereinigung, Durchschnitt, Komplement, kartesisches Produkt kann man wie folgt einführen:

- $\forall x \in X : \mu_{A \cup B} := \max\{\mu_A(x), \mu_B(x)\},$
- $\forall x \in X : \mu_{A \cap B} := \min\{\mu_A(x), \mu_B(x)\},$

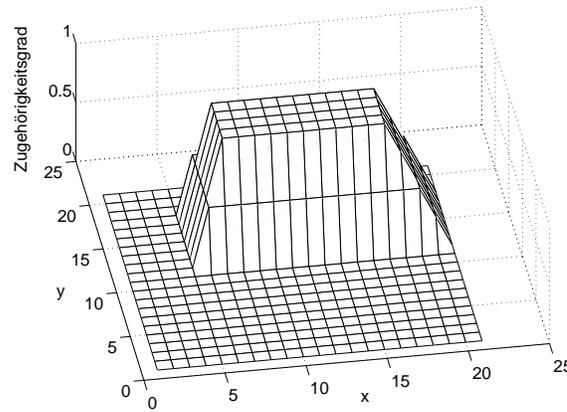


Abbildung 2.18: Eine 2-dimensionale trapezoide Zugehörigkeitsfunktion.

- $\forall x \in X : \mu_{\complement A} := 1 - \mu_A(x)$ ,
- $\forall x \in X : \mu_{A_1 \times \dots \times A_n}(x) := \min_i \{\mu_{A_i}(x_i) \mid x = (x_1, \dots, x_n), x_i \in X_i\}$ .

Es gibt allerdings weitere Möglichkeiten; erfüllt sein sollen dabei die gängigen Rechenregeln und einige Normierungseigenschaften.

Ergänzungen:

c) *Linguistische Variablen* können dazu dienen, sprachlich formulierte, z.B. medizinische, Sachverhalte zu modellieren. Eine **linguistische Variable**  $\mathcal{L}$  kann formal definiert werden als  $\mathcal{L} := (x, T(x), U, G, A)$  mit

- $x$ : Name der linguistischen Variablen  $\mathcal{L}$ ,
- $T(x)$ : linguistische Terme, d.h. eine Menge von Bezeichnern  $\{b_1, \dots, b_n\}$  von Fuzzy-Mengen  $\{B_1, \dots, B_n\}$ ,
- $U$ : Grundraum, über dem die Fuzzy-Mengen  $\{B_1, \dots, B_n\}$  definiert sind,
- $G$ : syntaktisches Regelwerk für die Bezeichner  $\{b_1, \dots, b_n\}$ , (i.d.R. eine formale Grammatik), um für den Menschen nur sinnvolle Bedeutungen zuzulassen und
- $A$ : semantisches Regelwerk, dass jedem Bezeichner  $\{b_1, \dots, b_n\}$  seine Bedeutung (mittels Zugehörigkeitsfunktionen  $\mu_{B_1}, \dots, \mu_{B_n} : U \rightarrow [0, 1]$ ) zuweist.

Beispiel: Die linguistische Variable  $\mathcal{L}$  in diesem einfachen Beispiel sei mit  $x = \text{Körpertemperatur}$  bezeichnet. Die linguistischen Terme sind  $T(x) = \{b_1, b_2, b_3\} := \{\text{erniedrigt, normal, erhöht}\}$ . Der Grundraum  $U$  ist durch  $U := [35, 42]$  gegeben. Exemplarisch wird  $A(\text{erniedrigt})$  definiert.  $A(\text{normal})$  und  $A(\text{erhöht})$  kann man analog definieren. Es sei  $A(\text{erniedrigt}) := \{(u, \mu_{\text{erniedrigt}}(u)) \mid u \in U\}$  mit  $\mu_{\text{erniedrigt}}(u) := 0, u \in [36.5, 42]$ ,  $\mu_{\text{erniedrigt}}(u) := \frac{u-36}{0.5}, u \in ]36, 36.5[$ ,  $\mu_{\text{erniedrigt}}(u) := 1, u \in [35, 36]$ . Das syntaktische Regelwerk  $G$  ist hier die Grammatik  $G = (S, N, T, P)$ , wobei  $S$  ein Startsymbol ist,  $N := \{S\}$  die Menge der

Nicht-Terminalsymbole,  $T := \{\text{erniedrigt, normal, erhöht}\}$  die Menge der Terminalsymbole und  $P := \{S \mapsto \text{erhöht}, S \mapsto \text{normal}, S \mapsto \text{erniedrigt}\}$  die Menge der Produktionsregeln.

d) **Modifikatoren** verändern die Bedeutung der Bezeichner, z.B. „stark“; man erhält anstelle von „erkältet“ „stark erkältet“.

e) Symbolische Variablen kann man *fuzzifizieren*, Schlüsse gemäß der vorhandenen *Fuzzy-Regeln* ziehen und die Ergebnisse *defuzzifizieren* (Schema z.B. wie in [NKK96, S. 268]).

- Fuzzifizierung: Umsetzen von Variablen in Fuzzy-Mengen,
- Fuzzy-Inferenz: Erstellen eines Regelwerkes mit Regeln der folgenden Art: „**if** stark erkältet **and** hohe Temperatur **then** starker Grippeverdacht“ und Finden neuer Regeln, alternativ: Neuro-Fuzzy-Inferenz: Lernen von Regeln mit einem RBF-Netz (Fuzzifizierung und Defuzzifizierung müssen im alternativen Fall entsprechend angepasst werden.)
- Defuzzifizierung: Umsetzen von Fuzzy-Ergebnismengen in (scharfe) Aussagen, z.B. nach der *Max-Min-Methode*, siehe [Zim91, S. 181-182].

f) (*Neuro-*)*Fuzzy-Regelung* führt etwas weiter, insofern man versucht mittels gelernter Regeln und Rückkopplungen ein System zu *regeln*, z.B. die (automatische) Blutzuckerregelung eines Patienten aufgrund seiner gemessenen Körperreaktion [Wic98].

**2.5.2 Beispiel:** Beispiel 1 (Fuzzy-Inferenz): CADIAG-2 ([Adl90] oder [Zim91, S. 185-189]) bzw. MEDFRAME/CADIAG-IV [B<sup>+</sup>96] zur Diagnoserstellung in der Inneren Medizin (z.B. Rheumatologie, Krankheiten des Pankreas und der Gallenblase).

Beispiel 2 (Neuro-Fuzzy-Inferenz): ANALYST zum Finden neuer Regeln, die der Klassifikation von Krankheiten dienen. Die Regelgenerierung basiert auf den im Abschnitt 2.2 vorgestellten RBF-DDA- bzw. RecBF-DDA-Netzen, siehe [BF00]. Hier wird insbesondere die didaktische Aufgabenstellung einer bedienerfreundlichen Benutzeroberfläche in der Medizin zur Regelvorgabe und -ausgabe beschrieben. Die o.g. Netze erzeugen deutlich zu viele Regeln im Vergleich zur Anzahl der Daten. [BF00] erhält für den dort verwendeten „Leber“-Datensatz mit 345 Trainingsvektoren (6 Variablen, 2 Klassen) 314 Regeln mit dem RBF-DDA-Netz und 86 Regeln mit dem RecBF-DDA-Netz; bei einem eigenen Versuch mit dem Fuzzy-RecBF-DDA-Netz wurden (im Mittel über 8 Versuchswiederholungen) nur 28 Regeln generiert. Außerdem unterliegt das erstgenannte RBF-DDA-Netz dem Problem der Abb. 2.16 Dennoch werden in [BF00] keinerlei weiterführende Regelgenerierungsverfahren verwendet und Güteabschätzungen durchgeführt. Dies wird in den folgenden Kapiteln im Rahmen der Analysen mit dem Fuzzy-RecBF-DDA-Netz nachgeholt.

Obwohl eine Regelgenerierung einen guten Zugang zu Analyseergebnissen bietet, müssen dafür einige Probleme überwunden werden:

- Vermeiden der Erzeugung zu komplizierter Regeln, die der Anwender nicht in seinen Begriffshorizont einbetten kann, d.h. Suche nach Kriterien zum Auffinden „einfacher“ Regeln,<sup>18</sup>
- Häufigkeits- und Konfidenz-Untersuchungen (siehe die Definitionen im Abschnitt 2.5.2 über Assoziations- und Generalisierungsregeln) zur Beurteilung der Qualität der gefundenen Regeln,
- gute Generalisierung, Vermeiden zu starker Spezialisierung,
- fehlende Werte und Ausreißer,
- übersichtliche Ausgabe für den Anwender, auch bei vielen Variablen und Regeln,
- Berücksichtigung des Regelgenerierungs-Klassifikations-Dilemmas ([NKK96], [CF99]),
- Interpretierbarkeit eines Neuro-Fuzzy-Systems [Nau00b, S. 39-40], d.h. nicht zu starke Überlappung der Gebiete, die durch die Regeln abgedeckt werden.

Der Vorteil neuronaler Ansätze zur Regelgenerierung liegt in der Verarbeitung beliebig komplex strukturierter Daten und der (heuristischen) Generalisierungsfähigkeit. Der Vorteil anderer nicht-neuronaler, „scharfer“ Verfahren ist die exakte Regelgenerierung, auch für symbolische Variablen.

Es werden einige weitere Ansätze aufgelistet, die hier nicht im Detail beschrieben werden und teilweise eine Regelgenerierung mit anderen Verfahren verbinden:

- sig\* zum unüberwachten Lernen neuer Regeln nach einer Clusterung mit einer SOM, siehe [Ult91, Abschnitte 7.5 und 7.6],
- basierend auf dem sig\*-Algorithmus und dem U-Matrix-Ansatz werden in [Gui98], [GU99] Regeln für multivariate Zeitreihenmuster generiert, um Apnoen zu klassifizieren,
- Idee der überwachten Regelgenerierung mit SGNG [Fri97b],
- NEFCLASS (Fuzzy-Backpropagation) [NK97], angewendet auf einen medizinischen Benchmark-Datensatz [Nau99],
- Regelextraktion für neuronale Netzwerke mit sigmoider Ausgabefunktion (z.B. FF-Backpropagation-Netz) durch Approximation mit Booleschen Funktionen [Tsu00a],

---

<sup>18</sup>Regeln können natürlich nicht einfacher sein als es die Daten ermöglichen.

- Integration von gesicherten, vorhandenen Regeln in Neuro-Fuzzy-Regelgeneratoren [Hol93],
- FUZZY-ARTMAP zur Klassifikation und Regelgenerierung (Lernen der Klassen Herzinfarkt, erhöhte Kreatininkinase, Tod) [LTL<sup>+</sup>99].

Weitere Neuro-Fuzzy-Systeme werden in [NKK96] und [Wit00, S. 120-162] beschrieben. Aufgrund der Verbreitung und der prinzipiell ähnlichen Leistungsfähigkeit von NEFCLASS im Vergleich zum Fuzzy-RecBF-DDA wird kurz auf die Vor- und Nachteile beider Verfahren eingegangen:

**Vorteile von NEFCLASS:** keine Regelgewichtung und somit eindeutig interpretierbare Regeln [Nau00a].

**Nachteile von NEFCLASS:** Vorgabe der Zugehörigkeitsfunktionen notwendig und somit auch deren Anzahl, langsamere Lernphase durch Backpropagation, jeweils alle Attribute gehen in die Regeln ein, zusätzliche Regelaggregation als Nachbearbeitung empfehlenswert, Verwendung von Kegel als Zugehörigkeitsfunktionen, d.h. die Core-Bereiche sind punktförmig und in ihrer Größe unveränderbar.

**Vorteile von Fuzzy-RecBF-DDA:** plausible Heuristik, vollständig adaptives Lernen ohne Vorgaben, keine Regelaggregation notwendig (da Regeln nur bei Notwendigkeit geschrumpft werden), irrelevante Attribute werden in den Regeln erkannt, Verwendung von Trapezoiden als Zugehörigkeitsfunktionen, d.h. unterschiedlich große Core-Bereiche sind möglich.

**Nachteile von Fuzzy-RecBF-DDA:** Ausreißerempfindlichkeit, Regelgewichtungen notwendig, stärkere Überlappungen der Regeln.

Für meine Analysen überwiegen die Vorteile von Fuzzy-RecBF-DDA, da wir keine Zugehörigkeitsfunktionen vorgeben wollen und später in der Tat rechteckige scharfe Regeln anstelle von Fuzzy-Regeln bevorzugen.<sup>19</sup> Das Erkennen irrelevanter Attribute sowie die Vermeidung zusätzlicher Regelaggregationen erscheint ebenfalls attraktiv.

Ergänzung: In [Wit00] wird bevorzugt NEFCLASS verwendet, was u.a. mit dem Argument der freien Verfügbarkeit von NEFCLASS begründet wird. Später werde ich das von mir implementierte, modifizierte Fuzzy-RecBF-DDA-Verfahren mit unendlichen Seitenausdehnungen aufgrund der genannten Vorteile verwenden. Allerdings würde ich in der Tat NEFCLASS als ein ähnlich gutes Verfahren einstufen, wenn man von vornherein Terme und deren Anzahl wie „hoch“ o.ä. vorgeben kann. – Im Unterschied zu der prinzipiell sinnvollen Vorgehensweise in [Wit00] werde ich auch Ergebnisse von *Versuchswiederholungen* wiedergeben, da ansonsten die Möglichkeit von zufällig guten oder schlechten Ergebnissen

---

<sup>19</sup>Eine grundsätzliche Diskussion über das Für und Wider von scharfen bzw. unscharfen Regeln wird in [DAG00] geführt.

nicht ausgeschlossen werden kann. Insbesondere ein Arzt ist auf statistisch abgesicherte Ergebnisse, z.B. durch Versuchswiederholungen, angewiesen. Dieser Zeitaufwand sollte aufgebracht werden.

## 2.5.2 Regelgenerierung im Maschinellen Lernen

Zwei prinzipiell mögliche Ansätze zum Auffinden von häufig vorkommenden Mustern und deren Abhängigkeiten in einem vorhandenen Datensatz sind zum einen die Generierung von *Assoziationsregeln* und zum anderen die Bildung von *Generalisierungsregeln*. Diese Verfahren werden hauptsächlich für symbolische Variablen eingesetzt, sind aber nicht notwendig darauf beschränkt. Es handelt sich um Ansätze aus dem Bereich des Maschinellen Lernens.

Eine formale Einführung bez. der Generierung von Assoziationsregeln wird in [Toi96, Kap. 2], ein Überblick über Algorithmen in [DXGH] gegeben. Ein einfacher und sehr bekannter Algorithmus ist der *A-priori-Algorithmus* [AS94], der in die beiden Schritte *Itemseterzeugung* und *Identifikation häufiger Itemsets* aufgeteilt ist. Eine Einführung bez. der Generalisierung gibt [Mit97, Kap. 2]. Die Grundlage der Verfahren bildet in dieser Arbeit eine Ansammlung von Mengen (engl.: *Itemsets*),<sup>20</sup> die jeweils elementare Attributausprägungen (engl.: *Items*) als Elemente besitzen. In einer Datenbanktabelle mit Attributen schließen sich verschiedene exklusive Attributausprägungen desselben Attributs aus; bei einer Implementierung sollte dies berücksichtigt werden, um nicht verbotene Kombinationen zu überprüfen. Die Definitionen sind zu diesem Spezialfall konsistent, da die Häufigkeit einer verbotenen Assoziationsregel im Abschnitt 2.5.3 c) immer 0 ist und somit die Regel nicht vorhanden ist.

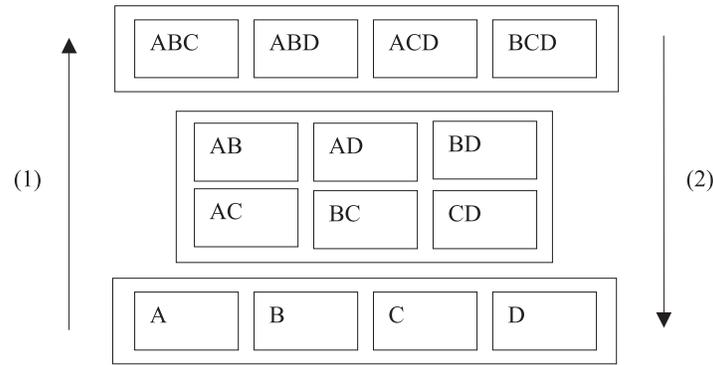
Im Folgenden sollen die beiden Paradigmen und ihre Zielsetzung formal eingeführt werden inklusive einiger wichtiger Bewertungsmaße für die Güte der Regelgenerierung. Die Definition und der englische Sprachgebrauch in der Def. 2.5.3 bezieht sich auf die Ideen in [AS94], die Def. 2.5.4 lehnt sich an die Def. 2.5.3 an. Der hier einheitlichen Darstellung wird zusätzlich die Definition der *gewichteten Konfidenz* hinzugefügt. In der Abb. 2.19 ist das Prinzip der (1) Assoziationsregeln und der (2) Generalisierungsregeln dargestellt. Zum ersten Mal werden die beiden Regelparadigmen einheitlich dargestellt. Die verwendeten Schreibweisen sind im Symbolverzeichnis zusammengestellt.

### 2.5.3 Definition: (Assoziationsregel)

Es seien elementare Attributausprägungen vorgegeben, sog. **Items**.  $I$ ,  $J$  und  $K$  seien Mengen von Items, sog. **Itemsets** (oder auch **Muster**), die in der Menge  $\mathcal{IS}$  zusammengefasst werden.  $\#$  sei im Folgenden die Anzahl der Elemente einer Menge.

---

<sup>20</sup>Die Definition der *Ansammlung* erfolgt in der Def. 4.1.1. Im Folgenden wird aber o.B.d.A. immer der Begriff *Menge* verwendet, auch wenn eine *Ansammlung* zugelassen ist, da später die Ansammlung auf eine Menge reduziert wird.



**Abbildung 2.19:** Prinzip der (1) Assoziationsregeln, (2) Generalisierungsregeln. Zu sehen sind die Mengen von Itemsets mit den Items A,B,C,D.

a) Die **Häufigkeit** (engl.: Frequency)  $freq(I)$  des Itemsets  $I \in \mathcal{IS}$  ist das Verhältnis der Anzahl derjenigen Itemsets  $K \in \mathcal{IS}$ , die das Itemset  $I$  enthalten, zur Gesamtzahl der Itemsets, d.h.

$$freq(I) := \frac{\#\{K \in \mathcal{IS} | I \subset K\}}{\#\mathcal{IS}} = \frac{\#\mathcal{IS}(I)}{\#\mathcal{IS}}. \quad (2.21)$$

Ist  $\min_{freq} \in [0, 1]$  eine untere Schranke für die Häufigkeit, so heißt  $I$  **häufig** (engl.: frequent), wenn  $freq(I) \geq \min_{freq}$  gilt.

b) Der Ausdruck  $I \Rightarrow J$  heißt **Assoziationsregel** (kurz: Regel) (bez. der Menge der Itemsets).

c) Die **Häufigkeit**  $freq(I \Rightarrow J)$  der Regel  $I \Rightarrow J$  ist definiert durch  $freq(I \cup J)$ . Aufgrund der Kommutativität des Vereinigungsoperators gilt  $freq(I \Rightarrow J) = freq(J \Rightarrow I)$ . Ist  $\min_{freq} \in [0, 1]$  eine untere Schranke für die Häufigkeit, so heißt die Regel  $I \Rightarrow J$  **häufig**, wenn  $freq(I \Rightarrow J) \geq \min_{freq}$  gilt. Mit  $Freq$  werde die Auflistung der häufigen Regeln bezeichnet.

d) Die **Konfidenz** (engl.: Confidence) einer Regel  $I \Rightarrow J$  wird definiert als die Anzahl derjenigen Itemsets  $K \in \mathcal{IS}$ , die die Items von  $I$  und  $J$  enthalten, geteilt durch die Anzahl derjenigen Itemsets  $K$ , die nur  $I$  enthalten:

$$conf(I \Rightarrow J) := \frac{\#\{K \in \mathcal{IS} | (I \cup J) \subset K\}}{\#\{K \in \mathcal{IS} | I \subset K\}} = \frac{\frac{\#\mathcal{IS}(I \cup J)}{\#\mathcal{IS}}}{\frac{\#\mathcal{IS}(I)}{\#\mathcal{IS}}} = \frac{freq(I \cup J)}{freq(I)}. \quad (2.22)$$

Ist  $\min_{conf} \in [0, 1]$  eine untere Schranke für die Konfidenz, so heißt  $I \Rightarrow J$  **konfident** (engl.: confident), wenn  $conf(I \Rightarrow J) \geq \min_{conf}$  gilt. Mit  $Conf$  werde die Auflistung der konfidenten Regeln bezeichnet.

e) Zielsetzung: In Abhängigkeit der Anwendung wähle man die Schranken  $\min_{freq}$  und  $\min_{conf}$ . Gesucht sind alle Assoziationsregeln  $I \Rightarrow J$ , so dass  $I, J \neq \emptyset, I \cap J = \emptyset, freq(I \Rightarrow J) \geq \min_{freq}$  und  $conf(I \Rightarrow J) \geq \min_{conf}$  gilt.<sup>21</sup>

Die Assoziationsregelmechanismen erzeugen die klassischen „Warenkorbregeln“ aus dem Bereich des kommerziellen Data Minings (vgl. [Bol96]): „Wenn Windeln gekauft, dann auch Bier gekauft.“ Ein Ladenbesitzer kann bei einer hohen Konfidenz dieser Regel die Waren Windeln und Bier nebeneinander stellen, was ihm u.U. einen Mehrumsatz einbringen kann und dem Kunden Laufarbeit spart, siehe aber dazu noch die Diskussion im Abschnitt 4.8. Ob diese Regel *tatsächlich* einen Nutzen bringt, ist nicht sicher, da das Optimierungsproblem „Positioniere alle Waren optimal, so dass ein maximaler Umsatz erzielt wird!“ nicht durch das Umstellen einzelner Waren gelöst wird, d.h. der Ladenbesitzer muss die durch die Assoziationsregeln generierte *Hypothese* überprüfen.

Weitere Anwendungen von Assoziationsregeln sind:

- Entdeckung von Alarmursachen in Telekommunikationsnetzen [HKM<sup>+</sup>96],
- Beschreibung von Herzkrankheiten [O<sup>+</sup>01],
- Wettervorhersage [Höp01] und
- Text-Mining in medizinischen Texten [BP01].

Wichtige Erweiterungen der „klassischen“ Assoziationsregeltechniken sind:

- Verwendung der Regeln zur Klassifikation [LHM98],
- Verwendung metrischer Daten [SA96],
- Integration von Einschränkungen (engl.: Constraints) des Suchraums in die Regelgenerierung [SVA97],
- Analyse von Zeitreihendaten [AS95],
- Berechnung von Interessantheitsmaßen für Regeln [HH99b],
- Sampling der Daten, um die Geschwindigkeit der Regelerzeugung zu erhöhen [Toi96, Kap. 5],
- Parallelisierung der Regelerzeugung [Zak98] und
- Visualisierung der Regeln [WWT99].

---

<sup>21</sup>Je nach Anwendung können weitere Gütemaße berücksichtigt werden, siehe Kap. 4.

Im Folgenden betrachten wir als Anwendung der Regeln die *Klassifikation*. Möchte man die Assoziationsregeln zur Klassifikation einsetzen, so beinhaltet das Itemset  $J$  die Klassenzugehörigkeiten  $\{K_1, \dots, K_m\}$ , aber keine allgemeinen Attribute.

Eine *Generalisierungsregel* wird nicht *bottom-up* erzeugt wie eine Assoziationsregel durch Bildung der Attributkombinationen, sondern man geht von den tatsächlich in der Potenzmenge der Items vorhandenen Itemsets aus und generalisiert diese *top-down*. Da man bei der Anwendung eines Assoziationsregelalgorithmus sehr viele, evtl. gar nicht vorkommende, Kombinationen aus den Items bilden muss [AS94], ist eine Idee bei einer Generalisierung nur Muster zu bilden, die tatsächlich vorkommen können (Kap. 4). Aber auch dieser Mechanismus kann kombinatorisch explodieren.

Ein wesentlicher Vorteil im Gebrauch der Generalisierung zur *Klassifikation* im Vergleich zur Assoziationsregelgenerierung ist die Möglichkeit der Verwendung einer Klasseninformation für ein Itemset (falls vorhanden); einzelnen Items kann keine Klasse zugeordnet werden. Man möchte möglichst pro Klasse  $K$  frequente und konfidente Regeln der Form  $I \Rightarrow K$  erzeugen. Bei den Assoziationsregeln kann man dies dadurch erreichen, indem man auf der Konklusionsseite der Regel nur die Klasseninformation zulässt. Eine Klassifikation, die auf einer heuristisch ausgewählten kleinen Regelmenge aus der i. Allg. sehr großen Gesamtmenge der Assoziationsregeln beruht, wird in [LHM98] beschrieben.

Um die Erzeugung von Generalisierungsregeln für die Klassifikation zu nutzen, werden die Definitionen bez. der Assoziationsregeln auf die Generalisierungsregeln übertragen. Die folgenden Definitionen werden losgelöst von einem konkreten Algorithmus gegeben. Sie werden allerdings so erweitert, dass konsequent eine nicht-eindeutige, „unscharfe“ Klassenzugehörigkeit verwendet werden kann. Zur algorithmischen Durchführung siehe dann das Kapitel 4. Die verwendeten Schreibweisen sind im Symbolverzeichnis zusammengestellt.

#### 2.5.4 Definition: (Generalisierungsregel)

$I$  und  $J$  seien Itemsets aus der Menge der Itemsets  $\mathcal{IS}$ .  $k^{(I)} = (k_1^{(I)}, \dots, k_m^{(I)})$  sei die *Klasseninformation* bez. aller  $m \in \mathbb{N}$  Klassen  $K_1, \dots, K_m$ , die für jedes Itemset gegeben ist, d.h.  $k_l^{(I)}$  beschreibt die Anzahl der Itemsets  $I$ , die zur Klasse  $K_l$  gehören. Ein Itemset muss hier also nicht genau einer Klasse zugehören, sondern nur mit einer bestimmten Häufigkeit  $k_l$ . Also kann man mittels der Klasseninformation  $m$  Regeln  $I \Rightarrow K_1, \dots, I \Rightarrow K_m$  gleichzeitig betrachten, d.h. der Ausgangspunkt sind bez. der Klassenzugehörigkeit „unscharfe“ Regeln.

a) Die **Häufigkeit** (engl.: Frequency)  $freq(I)$  des Itemsets  $I \in \mathcal{IS}$  ist identisch zu Def. 2.5.3 a) definiert.  $\min_{freq}$  wird genauso verwendet wie in der Definiton 2.5.3 a).

b) Gegeben seien  $m$  Itemsets  $I_1, \dots, I_m$ . Weiterhin seien  $p \leq m$  Regeln  $I_1 \Rightarrow K_*, \dots, I_p \Rightarrow K_*$  gegeben ( $*$  deutet jeweils beliebige Klassen an). Entsteht das Itemset  $L$  aus den Itemsets  $I_1, \dots, I_p$ , d.h.  $\emptyset \neq L \in \cap_l I_l, \forall l = 1, \dots, p : L \neq I_l$ , so heißt  $L \Rightarrow K_*$  **Generalisierungsregel** (kurz: Regel) (bez. der Menge der Itemsets) mit der neuen Klasseninformation  $k^{(L)}$ . Es wird vereinbart, dass die am Anfang vorhandenen Itemsets mit zur Menge der Generalisierungsregeln gehören.

c) Die **Häufigkeit**  $freq(I \Rightarrow K_l)$  der Regel  $I \Rightarrow K_l$  ist (im Unterschied zur Def. 2.5.3 c)) einfach definiert durch  $freq(I) = \#\mathcal{IS}(I)$ . Ist  $\min_{freq} \in [0, 1]$  eine untere Schranke für die Häufigkeit, so heißt die Regel  $I \Rightarrow K_l$  **häufig**, wenn  $freq(I \Rightarrow K_l) \geq \min_{freq}$  gilt. Mit  $Freq$  werde die Auflistung der häufigen Regeln bezeichnet.

d) Zusätzlich wird das Maß  $freq_{K_l}$  für die **Abdeckung** der Klasse  $K_l$  (oder  **$K_l$ -Häufigkeit**) definiert als die Summe der Klassenhäufigkeiten derjenigen Itemsets, die  $I$  enthalten, geteilt durch die Summe der Klassenhäufigkeiten aller Itemsets, aus denen (mind. einmal)  $K_l$  folgt:

$$freq_{K_l}(I \Rightarrow K_l) := \frac{\sum_{J \in \mathcal{IS}, I \subset J} k_l^{(J)}}{\sum_{J \in \mathcal{IS}, J \Rightarrow K_l} k_l^{(J)}} = \frac{\#\mathcal{IS}_l(I)}{\#\mathcal{IS}_l} . \quad (2.23)$$

Ist  $\min_{freq_{K_l}} \in [0, 1]$  eine untere Schranke für die *Abdeckung*, so heißt die Regel  $I \Rightarrow K_l$  **abgedeckt** oder **häufig bez. der Klasse  $K_l$** , wenn  $freq_{K_l}(I \Rightarrow K_l) \geq \min_{freq_{K_l}}$  gilt.

e) Sei  $I \Rightarrow K_l$  eine Generalisierungsregel. Die **Konfidenz** einer Regel  $I \Rightarrow K_l$  **bez. einer Klasse  $K_l$**  (kurz:  **$K_l$ -Konfidenz**) wird definiert als die Summe der Häufigkeiten der Itemsets  $J$  bez. der Klasse  $K_l$ , die  $I$  enthalten, geteilt durch die Summe der Häufigkeiten der Itemsets  $J$ , die  $I$  enthalten, bez. aller Klassen:

$$conf_{K_l}(I \Rightarrow K_l) := \frac{\sum_{J \in \mathcal{IS}, I \subset J} k_l^{(J)}}{\sum_{J \in \mathcal{IS}, I \subset J} \left( \sum_{s=1}^m k_s^{(J)} \right)} = \frac{\#\mathcal{IS}_l(I)}{\#\mathcal{IS}(I)} . \quad (2.24)$$

Analog werden *konfidente* Regeln bez. einer Klasse und die Menge  $Conf_{K_l}$  der konfidenten Regeln bez. der Klasse  $K_l$  definiert (unter Verwendung von Schwellen  $\min_{conf_{K_l}}$ ).

f) Da die Konfidenz bez. einer Klasse von der Datenaufteilung auf die verschiedenen Klassen abhängig ist, ist dieses Maß nicht gut geeignet, um Regeln bez. verschiedener Klassen zu vergleichen. Aus diesem Grund führe ich die **gewichtete Konfidenz bez. der Klasse  $K_l$**  ein als die  $K_l$ -Häufigkeit der Regel geteilt durch die Summe

der  $K_s$ -Häufigkeiten aller Klassen  $s$ :<sup>22</sup>

$$wconf_{K_l}(I \Rightarrow K_l) := \frac{freq_{K_l}(I \Rightarrow K_l)}{\sum_{s=1}^m freq_{K_s}(I \Rightarrow K_s)} \quad (2.25)$$

$$= \frac{conf_{K_l}(I \Rightarrow K_l) \frac{\#IS}{\#IS_l}}{\sum_{s=1}^m \left( conf_{K_s}(I \Rightarrow K_s) \frac{\#IS}{\#IS_s} \right)}. \quad (2.26)$$

Die Verwendung der gewichteten Konfidenz anstelle der (ungewichteten) Konfidenz macht immer dann Sinn, wenn die Klassenhäufigkeit nicht für alle Klassen gleich ist (wenn z.B. bei 2 Klassen also nicht 50% der Daten der Klasse 1 und 50% der Daten der Klasse 2 angehören), was in Anwendungen die Regel ist. Angenommen man hat 2 Klassen und 100 Samples (Klasse 1 gehören 80% der Daten an, der Klasse 2 nur 20%) und eine Regel  $I$ , die jeweils 15 Samples beider Klassen abdeckt, d.h.  $freq(I) = 30/100 = 0.3$ ,  $freq_{K_1}(I) = 15/20 = 0.75$  und  $freq_{K_2}(I) = 15/80 = 0.1875$ .

Dann ist die Konfidenz bez. beider Klassen gleich  $15/(15 + 15) = 0.5$ . Die Regel ist aber für die Klasse 1 eher uninteressant, da die triviale Regel „es folgt immer Klasse 1“ bereits die Konfidenz 0.8 hätte, aber für die Klasse 2 interessant (die triviale Regel „es folgt immer Klasse 2“ hätte nur die Konfidenz 0.2). Diesem Sachverhalt trägt die gewichtete Konfidenz Rechnung, die in diesem Beispiel für die Regel bez. der Klasse 1 einen Wert  $< 0.5$  und bez. der Klasse 2 einen Wert  $> 0.5$  hat; genau berechnet beträgt die gewichtete Konfidenz bez. der Klasse 1  $(15/80)/(15/80 + 15/20) = 0.1875/(0.1875 + 0.75) = 0.1875/0.9375 = 0.2$  und bez. der Klasse 2  $(15/20)/(15/80 + 15/20) = 0.75/(0.1875 + 0.75) = 0.75/0.9375 = 0.8$ .

Analog zu den konfidenten Regeln werden *gewichtskonfidente* Regeln bez. einer Klasse und die Menge  $WConf_{K_l}$  der gewichtskonfidenten Regeln bez. der Klasse  $K_l$  definiert (unter Verwendung von Schwellen  $\min_{wconf_{K_l}}$ ).

g) Zielsetzung: In Abhängigkeit der Anwendung wähle man die Schranken  $\min_{freq}$ ,  $\min_{conf_{K_l}}$ ,  $\min_{wconf_{K_l}}$  und  $\min_{freq_{K_l}}$ . Gesucht sind alle Generalisierungsregeln  $L \Rightarrow K_l$ , für die  $freq(L \Rightarrow K_l) \geq \min_{freq}$ ,  $conf_{K_l}(L \Rightarrow K_l) \geq \min_{conf_{K_l}}$ ,  $wconf_{K_l}(L \Rightarrow K_l) \geq \min_{wconf_{K_l}}$  und  $freq_{K_l}(L \Rightarrow K_l) \geq \min_{freq_{K_l}}$  gilt. Die Menge der Generalisierungsregeln, die die Auswahlkriterien erfüllen, sei mit *GenRule* bezeichnet.

Anwendungsbeispiel: In [BLH99] wird ein spezieller heuristischer hierarchischer Generalisierungsregel-Algorithmus eingeführt, allerdings ausschließlich für geordnete Attribut-Tupel und bez. einer Klasse „Missbrauch“ (engl.: Fraud), der bei der Aufdeckung von Kreditkartenmissbrauch eingesetzt wurde. Der Algorithmus wurde auch

<sup>22</sup>Ursprünglich habe ich die gewichtete Konfidenz durch die Konfidenz definiert; eine Umformung erleichtert aber die Definition.

mit einem Klassifikator für metrische Werte kombiniert, allerdings wurde kein Regelgenerator für metrische Werte eingesetzt. Die obigen Definitionen sind nur sinngemäß, aber nicht exakt auf die dortige Problemstellung übertragbar. Insbesondere wird durch obige Definition die Regelgenerierung und die Klassifikation für *mehrere* Klassen unterstützt.

Assoziationsregeln verwendet man, wenn man an *allen* häufigen Regeln in der Menge der Itemsets interessiert ist. In zumeist kommerziellen Anwendungen ähnlich der Warenkorbanalyse oder einer Kundendatenanalyse werden allerlei Varianten verwendet, die auf diesem Prinzip beruhen. Ein Problem ist die Komplexität der Algorithmen, die bei sehr großen Datenbanken (z.B. 1 Million Transaktionen, 1000 Items) besonders problematisch ist. Lösungsansätze hierfür sind Sampling (z.B. [Toi96, Kap.5] oder später den Abschnitt 4.6) oder Parallelisierung [Zak98]. – Generalisierungsregeln scheinen ein interessanter Ansatz zu sein, wenn eine Klassenzugehörigkeit existiert, die zudem auch *unscharf* sein kann. Dann entsprechen die Klasseninformationen den Konklusionsseiten der Regeln. Im Kap. 4 werden einige interessante Eigenschaften hergeleitet, die zeigen, dass eine einseitige Beschränkung auf Assoziationsregeln bei einer Klassifikationsaufgabe nicht sinnvoll ist. Die symbolischen Daten des septischen Schocks werden dann in den Kapiteln 6 und 7 durch den im Kap. 4 entwickelten Generalisierungsregelmechanismus klassifiziert.

Eine wichtiger Aspekt der durch die o.g. Definitionen grundsätzlich unterstützten Algorithmen ist vor allem die Laufzeit der konkreten Algorithmen, da sie kombinatorisch explodieren. In [AS94] werden die gebildeten Assoziationsregeln während des laufenden Verfahrens entfernt, wenn sie nicht mehr häufig genug werden können. In [HPY00] wird für die Generierung der häufigen Muster ein Präfix-Baum verwendet. In [PB01] wird solch ein Baum für die Regelgenerierung der ausgewählten symbolischen Klinikdaten eingesetzt, siehe den Abschnitt 6.6. – Bez. der Generalisierung wird aufgrund der kombinatorischen Explosion von einer optimalen Generalisierung abgewichen, die nur auf einem leistungsstarken (Parallel-)Rechner möglich wäre. Für die Generalisierungsregeln wird eine Heuristik entwickelt, die die kombinatorische Explosion vermeidet, indem nicht alle Durchschnitte gebildet werden und ein evtl. suboptimales Ergebnis in Kauf genommen wird (siehe den Abschnitt 4.6).

Ergänzung: Eine weitere Variante der Regelgenerierung sind *Entscheidungsbäume* ([Mit97, Kap. 3], [BH99, Kap. 6], [KWZ98], [AZ96]). Ein *Entscheidungsbaum* ist hierarchisch aufgebaut. An den Knoten wird je nach Wert einer Variablen verzweigt (z.B. nach einem Informationskriterium wie dem *Informationsgewinn*, „Blutdruck < 100“ linker Ast, „Blutdruck  $\geq$  100“ rechter Ast). Der Weg von der Wurzel zum Blatt repräsentiert die Regel, d.h. am Blatt wird den Daten eine Klasse zugeordnet. Je höher im Baum verzweigt wird, desto mehr Informationen liefert die Verzweigung der Variable, siehe z.B. das C4.5-Verfahren von Quinlan ([Qui93], [Mit97, Kap. 3]). Entscheidungsbäume haben den Vorteil, dass sie durch ihre hierarchische Heuristik ein gutes Laufzeitverhalten besitzen. Nicht gut verwendbar sind Entscheidungsbäume, wie bereits auf der S. 47 geschildert, in kompliziert strukturierten, hochdimensionalen Datensätzen.

Trotz der eigenständigen Bedeutung der *evolutionären* und *genetischen* Algorithmen als Optimierungsstrategien (z.B. [BH99, Kap. 9] oder [Gol89]), möchte ich nur kurz ein Beispiel aufführen, in dem sie als ergänzende heuristische Strategie zur (symbolischen) Regelgenerierung Verwendung finden. Die prinzipielle Funktionsweise liegt in der Anwendung der Mutations- und Selektionsoperatoren sowie der Fitness-Abbildung auf die Regeln. In [FLF00] wird ein genetischer Algorithmus zur Regelgenerierung aus Itemvektoren angewendet, dem als Fitnessfunktion das Produkt aus Sensitivität und Spezifität vorgegeben wurde.

Einige Anwendungen der Regelgenerierung auf medizinischen Datenbasen, die auf der sog. *Rough Set Theory* [Paw82] beruhen, werden in [Tsu99a] und [Tsu99b] beschrieben. Neben dem Attribut-basierten Lernen (Assoziations- und Generalisierungsregeln) werden im Teilgebiet der *Induktiven Logischen Programmierung* (Abk.: ILP, engl.: Inductive Logic Programming) auch Relationen zwischen den Attributen zum Lernen der Regeln (auf PROLOG-Basis) mitverwendet, siehe z.B. [BM95]. Ältere Verfahren wie „candidate elimination“ [Mit97, Kap. 2], die nicht mit widersprüchlichen Daten umgehen können oder spezielle logikbasierte Verfahren wie das *nichtmonotone Schließen* [BKI00, Kap. 7 und 8] werden hier ebenfalls nicht betrachtet.

### 2.5.3 Kombination von Regelmengen

In den beiden vorhergehenden Abschnitten wurden die Grundlagen der metrischen und der symbolischen Regelgenerierung vorgestellt. Der Vorteil der auf die jeweilige Datenart spezialisierten Verfahren ist die individuelle Anpassung an die metrischen bzw. symbolischen Daten, z.B. verwendet das Verfahren Fuzzy-RecBF-DDA die Möglichkeit zur Berechnung von Abständen der metrischen Daten, die zu beliebigen, kontinuierlichen Ausdehnungen der Regeln führt. Die Assoziations- bzw. Generalisierungsregeln arbeiten kombinatorisch auf den symbolischen, diskreten Daten. So erhält man Verfahren, die für die jeweilige Datenart einen passenden Auswertungsmechanismus besitzen.

Grundsätzlich ist es möglich, metrische Daten durch Partitionierung des Datenraums in symbolische Daten zu transformieren, in dem einem metrischen Wert der symbolische Wert einer Partition (z.B. eine Nummer) zugewiesen wird. Ein einfaches Beispiel ist die Rundung reeller Werte auf ganzzahlige Werte. Dadurch entsteht zwar eine Ungenauigkeit, aber es wird prinzipiell möglich, die transformierten metrischen Daten als symbolische aufzufassen und mit einem auf symbolische Daten spezialisierten Verfahren zu verarbeiten. Beispielsweise wurden im Abschnitt 4.4 die metrischen IRIS-Daten durch das Generalisierungsregelverfahren verarbeitet. Dieses Prinzip wird auch beim Entscheidungsbaum oder bei Assoziationsregeln verwendet. Der entscheidende Schritt ist dabei die sinnvolle Partitionierung, die z.B. durch eine statische Einteilung, durch eine Clusterung oder auch dynamisch entstehen kann.

Symbolische Daten, die einer mathematischen Ordnung unterliegen, können als metrische Daten aufgefasst werden. Da speziell das Fuzzy-RecBF-DDA-Netz für kontinuierliche Werte geeignet ist, sollten diese symbolischen Daten künstlich verrauscht werden, um die Diskretheit der Daten abzuschwächen, vgl. die Ausführungen auf der S. 74.

Symbolische Daten, die keiner Ordnung unterliegen (z.B. medizinische Operationen) können nicht mit einem metrischen Verfahren (insbesondere dem Fuzzy-RecBF-DDA-Netz) verarbeitet werden.

Insgesamt folgt, dass es wünschenswert wäre, die Vorteile der metrischen und symbolischen Regelgeneratoren beizubehalten und die Ergebnisse der Verfahren zu *kombinieren*, sowohl zum Erhalt von Regeln als auch zur kombinierten Klassifikation. Insbesondere hofft man auf eine bessere Klassifikationsleistung durch ein kombiniertes System als durch die jeweiligen Einzelsysteme.

Im Hinblick auf die Anwendung soll im Folgenden insbesondere die Kombination einer durch das Fuzzy-RecBF-DDA-Verfahren erzeugten metrischen Regelmenge  $R$  mit einer durch den Generalisierungsregelmechanismus erzeugten Regelmenge  $S$  diskutiert werden. Die Regelmenge  $R$  entstand aus metrischen Messungen am Patienten;  $S$  wurde aus den symbolischen Operations-, Therapie- und Medikamentenvariablen gebildet.

Ergänzung: Die Kombination von Klassifikationssystemen, die auf *denselben* Daten erzeugt wurden, im Sinne eines *Resamplings*, wurde kurz im Abschnitt 2.3.2 angesprochen.

Es wäre zunächst auch denkbar, verschiedene symbolische Regelmengen  $S_1, \dots, S_m$  untereinander oder metrische Regelmengen  $R_1, \dots, R_n$  untereinander zu kombinieren. Sinnvoll ist das, wenn ein Verfahren nur eine maximale Anzahl von Variablen gleichzeitig verarbeiten kann. Für die symbolischen Daten wäre z.B. ein System für die Operationen, eins für die Therapien und eins für die Medikamente denkbar. Für die metrischen Daten wäre z.B. eine Regelmenge pro Subsystem (Herz, Lunge, Niere usw.) denkbar. Wenn es möglich ist, sollte aber keine Kombination erfolgen, da ja die Multidimensionalität der Daten so nicht mehr vollständig vorhanden ist und ein hierarchisches System entsteht.

Im Fall der ASK- bzw. Klinikdaten können alle symbolischen Variablen gleichzeitig verarbeitet werden, so dass die Kombination nicht notwendig ist. Das Fuzzy-RecBF-DDA-Verfahren wird nach den gemachten Erfahrungen nicht mehr als etwa 20 metrische Variablen gleichzeitig sinnvoll verarbeiten können, so dass hier eine Kombination prinzipiell sinnvoll wäre. Allerdings ist sie hier nicht sinnvoll durchzuführen, da zu viele fehlende Werte die Kombination unmöglich machen: Für ein einzelnes Subsystem (sagen wir für die Argumentation mit einheitlich 10 Variablen)

sind etwa 6 bis 9 Werte gleichzeitig gemessen, was toleriert werden kann durch eine fehlende Werte-Ersetzung. Möchte man z.B. 5 Subsysteme kombinieren, so wären von  $5 \cdot 10 = 50$  benötigten Werten als Eingabe, die dann auf die 5 Subsysteme verteilt würden, oft nur etwa 20 Variablen gleichzeitig gemessen; so viele fehlende Werte kann man aber nicht mehr tolerieren. Aus den genannten Gründen wird an dieser Stelle diese Art der Kombination nicht separat diskutiert; allerdings sind die folgenden Ausführungen für die Kombination von  $R$  und  $S$  auch auf diese Situation sinngemäß übertragbar.

**2.5.5 Vereinbarung:** Sei also im Folgenden  $R$  eine metrische Regelmenge, die mit Fuzzy-RecBF-DDA aus den metrischen Daten  $M$  erzeugt worden ist. Für jeden Eingabevektor  $x^{(m)} \in M$  sei die prozentuale Ausgabe  $(p_1, p_2)$  gegeben (für zwei Klassen, vgl. den Abschnitt 7.5.1),<sup>23</sup> oder zusammengefasst  $(x^{(m)}; p_1, p_2)$ .

$S$  sei eine symbolische Regelmenge, die mit den Generalisierungsregeln aus den symbolischen Daten  $N$  erzeugt worden ist. Für jeden Eingabevektor  $x^{(s)} \in N$  sei  $(g_1, g_2)$  die gewichtete Konfidenz der am besten zutreffenden Regel für die jeweilige Klasse (kurz:  $(x^{(s)}; g_1, g_2)$ ), vgl. den Abschnitt 4.3.

Kann man jedem  $x^{(m)} \in M$  ein  $x^{(s)} \in N$  zuordnen (dann ist notwendig die Anzahl der Samples aus  $M$  gleich der aus  $N$ ), so sei  $x = (x^{(m)}, x^{(s)})$ .<sup>24</sup> Diese Eigenschaft sei hier als *Tupeleigenschaft* bezeichnet.

Ist die Tupeleigenschaft erfüllt, kann man als einfache Möglichkeit zur Kombination von Regeln das Kreuzprodukt  $T = R \times S$  bilden und die Gütemaße (z.B. Häufigkeit, Konfidenz) für die Regeln aus  $T$  berechnen. Der Nachteil ist die hohe Anzahl kombinierter Regeln, z.B. erhält man für  $\#R = \#S = 30$  Regeln bereits  $\#T = 900$  Regeln. Eine Möglichkeit der Reduzierung der Anzahl der Regeln wäre das nachträgliche Pruning von Regeln, die in der Kombination eine geringe Häufigkeit und/oder Konfidenz haben.

Unter der Voraussetzung, dass die Tupeleigenschaft erfüllt ist, kann man die Datensamples  $x$  aus der Trainingsmenge in  $x^{(m)} \in M$  und  $x^{(s)} \in N$  aufteilen und getrennt die Regelmengen erzeugen. Anschließend kann man die Testdaten  $x$  in  $x^{(m)} \in M$  und  $x^{(s)} \in N$  aufteilen und separat die Ausgaben gemäß der Vereinbarung berechnen. Zur Klassifikation könnte man mit den Ausgaben (d.h. den Prozentanteilen bzw. gewichteten Konfidenzen), z.B. mit einem neuronalen Netz, eine Klassifikation trainieren.

Sei speziell  $y_k^{(m)}$  die Ausgaben der Klasse  $k$  des Fuzzy-RecBF-DDA-Verfahrens und  $y_k^{(s)}$  die Ausgaben der Klasse  $k$  des Generalisierungsregel-Verfahrens. Mit einem weiteren Fuzzy-RecBF-DDA-Netz ließen sich zusätzlich zu der Klassifikation (mit

<sup>23</sup>O.B.d.A. können mehr als zwei Klassen betrachtet werden.

<sup>24</sup>Dies wird in unserer Anwendung aber so nicht der Fall sein.

$q_i \in [0, 100] \cup \{-\infty, +\infty\}$  als prozentuale Ausgabeanteile und  $h_i \in [0, 1] \cup \{-\infty, +\infty\}$  als gewichtete Konfidenzwerte) Regeln der Form:

„**if**  $y_1^{(m)}$  hat prozentualen Anteil **in**  $[q_1, q_2]$  **and**  
 $y_2^{(m)}$  hat prozentualen Anteil **in**  $[q_3, q_4]$  **and**  
 $y_1^{(s)}$  hat gewichtete Konfidenz **in**  $[h_1, h_2]$  **and**  
 $y_2^{(s)}$  hat gewichtete Konfidenz **in**  $[h_3, h_4]$   
**then** Klassenzugehörigkeit  $K$   
**with** Gütemaße (Häufigkeit und Konfidenz)“

erzeugen. Insbesondere kann so festgestellt werden, welche Kombination der Ausgabewerte der Regelmengen für die Klassenentscheidung verantwortlich ist und ob symbolische und/oder metrische Daten wichtiger für die Klassenentscheidung sind.

Im Abschnitt 7.4 wird speziell für die metrischen und symbolischen MEDAN-Daten eine Kombination der Regelmengen zur Klassifikation durchgeführt. Dort soll auch auf die speziellen Probleme, z.B. das Fehlen der Tupeleigenschaft, und auf die daraus folgenden Konsequenzen eingegangen werden.

In den folgenden Kapiteln wird anknüpfend an den Ausführungen des Abschnitts 2.5 ein Schwerpunkt auf die Regelgenerierung gelegt.

# Kapitel 3

## Regelgenerierung mit Fuzzy-RecBF-DDA-Netzen für metrische Daten

Im Anschluss an den Abschnitt 2.2.3 wird im Abschnitt 3.1 die DDA-Architektur mit einer Fuzzy-Aktivierungsfunktion kombiniert [HB95], um zusätzlich zur Klassifikation Regeln zu generieren. In Erweiterung des Abschnitts 2.5.1 wird auf die Schwächen dieses Verfahrens bezüglich der Regelgenerierung und der Klassifikation hingewiesen. Grundlegende Schwächen, die der Algorithmus besitzt und bislang nicht berücksichtigte Situationen, werden bereits in der Darstellung des Algorithmus verbessert und kenntlich gemacht. Experimente auf Benchmark-Daten werden im folgenden Abschnitt durchgeführt; insbesondere werden auch umfangreiche Konfidenz- und Häufigkeits-Untersuchungen für die einzelnen Regeln durchgeführt, die in diesem Umfang bislang noch nicht vorgenommen wurden. Beispielsweise wird im Abschnitt 3.4 die Konfidenz und die Häufigkeit in Abhängigkeit vom Fuzzy-Niveau untersucht, womit eine Transformation der Fuzzy-Regeln in scharfe, rechteckige Regeln verbunden ist.

Anschließend wird im Abschnitt 3.5 eine Kenngröße für die globale Wichtigkeit von Variablen eingeführt, mit deren Hilfe eine Dimensionsreduktion möglich ist. Auch wird die Frühzeitigkeit einer Regel in Hinblick auf eine medizinische Alarmgebung durch erzeugte Regeln definiert. Im Abschnitt 3.6 wird eine Variante des Fuzzy-RecBF-DDA-Verfahrens vorgestellt, die weniger empfindlich auf „Ausreißer“ reagiert.

Problematisch in einer Anwendung ist die Auswahl eines Modells aus vielen, insbesondere bei nicht repräsentativen Datensätzen, so dass im Abschnitt 3.7 auf der Basis von Ähnlichkeitsmaßen eine Möglichkeit vorgeschlagen wird, eine quantitativ gestützte Modellwahl zu treffen. In den Kapiteln 5 und 7 werden die Algorithmen dann auf die MEDAN-Daten angewendet.

Insgesamt bietet der grundlegende Algorithmus [HB95] interessante Eigenschaften (siehe den grundsätzlichen Vergleich mit NEFCLASS im Abschnitt 2.5.1 und die folgende Auflistung der Eigenschaften im Abschnitt 3.1), so dass er als Kandidat für die Anwendung auf den MEDAN-Daten in den Kapiteln 5 und 7 ausgewählt wurde. Auch konnten ausgehend vom o.g. Algorithmus sinnvolle Verbesserungen und Ergänzungen vorgenommen werden, die zu guten Ergebnissen führten. Andere Neuro-Fuzzy-Algorithmen haben grundsätzliche Nachteile, z.B. verwenden sie nicht die Klasseninformation der Daten und sind ausschließlich Cluster-basiert, oder man muss a-priori Wissen über die Zugehörigkeitsfunktionen oder über die Regeln vorgeben, vgl. Abb. 2-22 in [Wit00, S. 130].

### 3.1 Das Fuzzy-RecBF-DDA-Netz und Verbesserungen

Die modifizierten Algorithmen, die auf dem Verfahren Fuzzy-RecBF-DDA beruhen, werden vorgestellt. Die Verbesserungen werden im Anschluss an die Algorithmen explizit genannt. Die Funktionsweise wird veranschaulicht und grundsätzliche Vor- und Nachteile des Verfahrens werden gegenübergestellt. Dem Fuzzy-RecBF-DDA-Netz liegt die Netztopologie der Abb. 2.4 zugrunde. Zuerst werden die Parameter des Verfahrens vorgestellt.

Parameter:

Im Gegensatz zum RecBF-DDA werden hier pro Regel *zwei* Rechtecke adaptiert, ein inneres (*core*) und ein äußeres (*support*), so dass sich nach dem Lernprozess in kanonischer Weise trapezoide Zugehörigkeitsfunktionen und damit Fuzzy-Regeln erzeugen lassen. Für die Ausdehnungen der Rechtecke werden die Parameter  $\lambda_{d,i,-}^k, \lambda_{d,i,+}^k$  (inneres Rechteck) und  $\Lambda_{d,i,-}^k, \Lambda_{d,i,+}^k$  (äußeres Rechteck), Dimension  $d = 1, \dots, n$ , Anzahl der Regeln pro Klasse  $i = 1, \dots, m_k$  eingeführt. Optional wird der *nearby-Parameter* eingeführt, der weiter unten – zusammen mit weiteren grundsätzlichen Modifikationen – beschrieben wird.

Ergänzung (MATLAB-Datenstruktur für die Regel-Prototypen): Cell-Array mit Structs für jede Klasse  $k$  mit folgenden Parametern: `prototypes{k}.m` (Anzahl der Prototypen bez. Klasse  $k$ ), `prototypes{k}.z(i,:)` ( $i$ -ter Gewichtsvektor für den  $k$ -ten Prototypen,  $i = 1, \dots, m$ , „:“ steht für die Dimensionen), `prototypes{k}.w(i)` ( $i$ -ter Gewichtszähler), `prototypes{k}.lambda(i,:)` ( $i$ -te innere „Radien“ bzw. Ausdehnungen),  $\lambda_{d,i,\pm}^k$ , „:“ steht für die Dimensionen und die Richtung + bzw. –), `prototypes{k}.Lambda(i,:)` (äußere Radien) und `prototypes{k}.Kinfy(i,:)` (Unendlichkeitsanzeiger: 1 = unendlich und 0 = endlich, „:“ steht für die Dimensionen und die Richtung + bzw. –).

**Lernphase einer Epoche:** ähnlich zum RecBF-DDA mit dem Unterschied, dass die inneren und äußeren Rechtecke angepasst werden müssen. Das verwendete modifizierte shrink-Modul und die eigenen Modifikationen werden im Anschluss an den Algorithmus beschrieben. Der Originalalgorithmus ist für den interessierten Leser im Anhang D beigefügt.

### 3.1.1 Algorithmus (Fuzzy-RecBF-DDA, modifiziert):

Die Schritte beschreiben den Lernvorgang einer Epoche. Auf der S. 73 wird eine Benchmark-Umgebung für den Algorithmus vorgestellt.

1. Gewichte zurücksetzen:

```

for l = 1 to c
  for i = 1 to ml
    wil := 0;
    λd,i,± := 0;
  end
end

```

2. Muster  $x$  durchgehen ( $x$  sei aus der Klasse  $k$ ):

```

for r = 1 to # $\mathcal{DS}$  (durchlaufe Datensamples)
  if ( $p_i^k$  covers  $x$ ) and ( $\|z_i^k - x\|_2 < nearby$ ) then

```

3. Muster durch  $p_i^k$  überdeckt (cover):

```

  wik := wik + 1;
  adjustiere die λd,i,±k, so dass sie  $x$  überdecken;
  Gibt es einen Core-Konflikt, d.h. liegt  $x$  in einem Core-Bereich
  eines Prototypen einer anderen Klasse  $l \neq k$ , so setze alle
  λd,i,±k := 0; (kann passieren, wenn ein Core-Bereich
  von einem Support-Bereich einer anderen Klasse überlappt wird)

```

4. Neues Neuron einfügen (commit):

```

else
  zmk+1k := x;
  λd,mk+1,± := 0;   Core-Bereich ist nur der Zentrumspunkt
  Λd,mk+1,±k := ∞;   Support-Bereich ist unendlich

```

5. **for**  $l \neq k$ ,  $1 \leq j \leq m_l$

Schrumpfe  $p_{m_k+1}^k$  anhand  $z_j^l$ , d.h.

```

  shrink( $p_{m_k+1}^k, z_j^l$ );

```

```

end

```

```

  mk := mk + 1;

```

```

  wmkk := 1.0;

```

```

end

```

6. Konfliktneuronen verkleinern (shrink):

```

for  $l \neq k, 1 \leq j \leq m_l$ 
  if  $x$  liegt im Support-Bereich von  $p_j^l$  then neue, zusätzliche Abfrage
    Schrumpfe  $p_j^l$  anhand  $x$ , d.h.
    shrink( $p_j^l, x$ );
  end
end
end

```

Das Paradigma des Algorithmus 3.1.2 ist das Schrumpfen einer Regel in einer Dimension, so dass das Volumen der entstehenden Regel so groß wie möglich bleibt („Prinzip des minimalen Volumenverlustes“) ohne die Regeln in einigen Dimensionen zu schmal werden zu lassen. Auch hier wurden eigene Ergänzungen vorgenommen, die weiter unten beschrieben werden.

### 3.1.2 Algorithmus (Shrink-Mechanismus, modifiziert):

Shrink-Algorithmus shrink( $p, x$ ) („minimaler Volumenverlust“):

$M := |z_{\tilde{d}} - x_{\tilde{d}}|$  mit  $\tilde{d} := \text{index} \left( \min_{1 \leq d \leq n} \left\{ \left| \frac{\Lambda_{d,\pm} - |z_d - x_d|}{\Lambda_{d,\pm}} \right| \right\} \right)$ ;

(Berechne  $\tilde{d}$  mittels +-Richtung in der Dimension  $d$  für  $\Lambda$ , falls  $z_d - x_d \geq 0$ , sonst verwende die --Richtung, für alle *endlichen* Radien  $\Lambda$ .)

Falls  $M$  existiert, setze:

$\Lambda_{d,\text{min},\pm} := M$ ;

$\Lambda_{d,\text{bestfinite},\pm} := M$ , falls  $M \geq \sigma_{d,\text{min}}$ ;

$N := \max\{|z_d - x_d| \mid 1 \leq d \leq n\}$ ;

(Berechne  $N$  mittels +-Richtung in der Dimension  $d$  für  $\Lambda$ , falls  $z_d - x_d \geq 0$ , sonst verwende die --Richtung, für alle *unendlichen* Radien  $\Lambda$ .)

Falls  $N$  existiert, setze:

$\Lambda_{d,\text{max},\pm} := N$ ;

$\Lambda_{d,\text{bestinfinite},\pm} := N$ , falls  $N \geq \sigma_{d,\text{min}}$ ; neuer Wert für neue Fallunterscheidung

Berechne ein neues  $\Lambda_{d,\pm}$ , d.h. ein *shrink* in einer Richtung für einen Radius:

**if**  $\Lambda_{d,\text{bestfinite},\pm}$  existiert **then**

$\Lambda_{d,\pm} := \Lambda_{d,\text{bestfinite},\pm}$ ;

**else** neue Fallunterscheidung

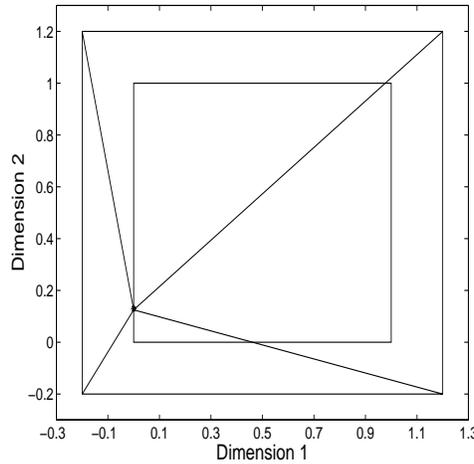
**if**  $\Lambda_{d,\text{bestinfinite},\pm}$  existiert **and**

(( $\Lambda_{d,\text{bestinfinite},\pm} \geq \Lambda_{d,\text{min},\pm}$ ) **or** ( $\Lambda_{d,\text{min},\pm}$  existiert nicht)) **then**

$\Lambda_{d,\pm} := \Lambda_{d,\text{bestinfinite},\pm}$ ;

**else**

**if**  $\Lambda_{d,\text{min},\pm}$  existiert **then**



**Abbildung 3.1:** Einfügen des ersten normierten Punktes  $(0.00, 0.13; 1)$ . Die Support-Region ist unendlich und die Core-Region ist nur der Punkt selbst. Man sieht eine zweidimensionale Projektion einer asymmetrischen Pyramide. Das kleinere Einheitsquadrat ist das „Endlichkeitsrechteck“, das die maximale Ausdehnung 1 andeutet. Das größere Quadrat (Seiten bei  $-0.2$  bzw.  $1.2$ ) deutet auf unendliche Ausdehnungen  $\infty$  bzw.  $-\infty$  hin.

```

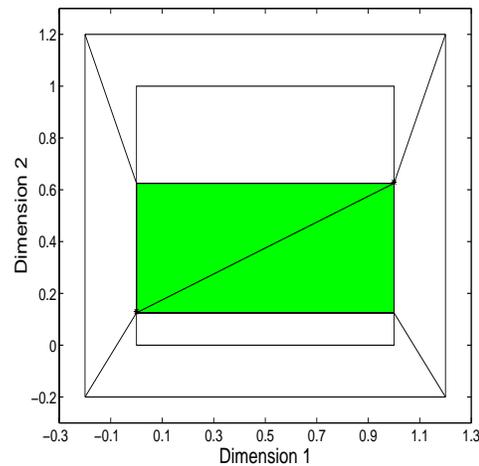
     $\Lambda_{d,\pm} := \Lambda_{d,\min,\pm};$ 
  else
     $\Lambda_{d,\pm} := \Lambda_{d,\max,\pm};$ 
  end
end
end
end

```

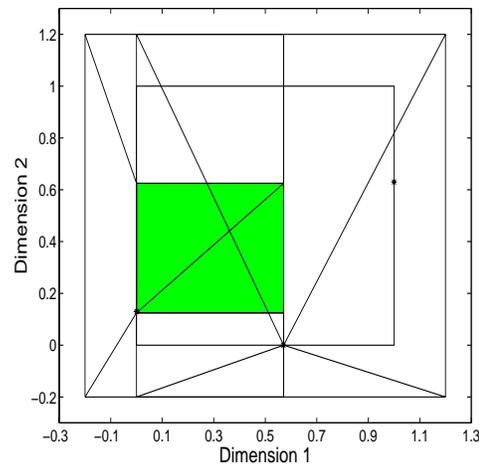
Abweichend zum Original-Algorithmus [HB95] wurden die folgenden sechs Ergänzungen vorgenommen:<sup>1</sup>

1. Unterscheidung zwischen dem neuen Wert  $\Lambda_{d,\text{bestinfinite},\pm}$  und  $\Lambda_{d,\max,\pm}$ , weil  $\Lambda_{d,\min,\pm}$  nicht immer existiert und diese Situation, dass für alle  $d = 1, \dots, n$  der Wert  $\Lambda_{d,\min,\pm}$  nicht existiert, nicht berücksichtigt wurde, vgl. hierzu auch die Ausführungen im Anhang D auf der S. 293.
2. Schrumpfung der Radien  $\lambda$ , falls gilt:  $\Lambda < \lambda$  (für ein jeweiliges  $d, i, \pm$ ), d.h. hat die Support-Region nach Schrumpfung des Radius in einer Dimension einen kleineren Radius als die der Core-Region, so muss auch die Core-Region entsprechend geschrumpft werden (nicht extra erwähnt im Alg. 3.1.2).
3. Modul, das verhindert, dass zwei Core-Bereiche zweier *verschiedener* Klassen sich überlappen. Die Punkte 2. und insbesondere 3. verhindern eine nicht erwünschte Überlappung von Core-Bereichen (*Core-Konflikt*) bei der Regelgewinnung z.B. aus den Daten (dim1,dim2;Klasse):  $(0.0, 0.3; 1)$ ,  $(0.7, 0.7; 1)$ ,

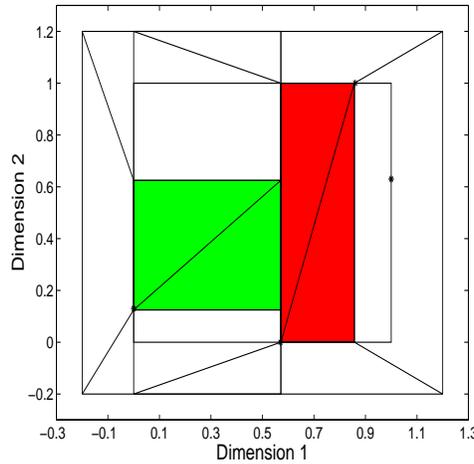
<sup>1</sup>Der Originalalgorithmus kann zum Vergleich im Anhang D eingesehen werden.



**Abbildung 3.2:** Einfügen des zweiten normierten Punktes  $(1.00, 0.63; 1)$ . Die Support-Region ist unendlich und die Core-Region ist nun das hellgraue Rechteck. Der erste Punkt ist das Zentrum der Regel. Vom Zentrum ausgehend sind Verbindungslinien zu den Eckpunkten der Core-Region eingezeichnet.



**Abbildung 3.3:** Einfügen des dritten normierten Punktes  $(0.57, 0.00; 2)$ . Die Support-Region der ersten Regel ist geschrumpft; damit auch die Core-Region. Die zweite Regel hat als Support-Region den bez. der ersten Regel bereits geschrumpften Bereich und als Core-Region nur den Punkt selbst.

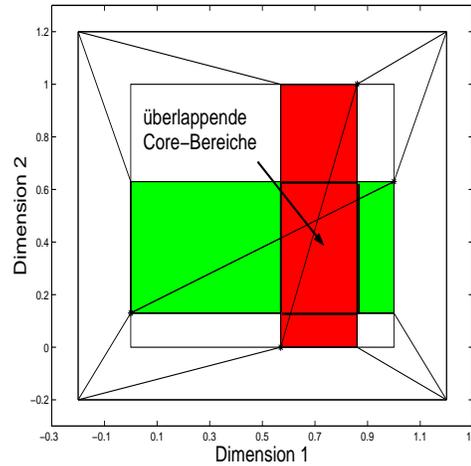


**Abbildung 3.4:** Einfügen des vierten normierten Punktes  $(0.86, 1.00; 2)$ . Die Support- und Core-Region der ersten Regel wird nicht mehr verändert. Die zweite Regel hat als Core-Region nun das rechte „horizontale“ dunkle Rechteck.

$(0.4, 0.2; 2)$ ,  $(0.6, 1.0; 2)$ , die normiert in den Dimensionen zwischen 0 und 1 eingefügt werden:  $(0.00, 0.13; 1)$ ,  $(1.00, 0.63; 1)$ ,  $(0.57, 0.00; 2)$ ,  $(0.86, 1.00; 2)$ . Der Ablauf wird, auch zur generellen Anschauung des veränderten Verfahrens, in den Abbildungen 3.1, 3.2, 3.3 und 3.4 dargestellt.<sup>2</sup> Im Originalverfahren hätte man zwei überlappende Core-Bereiche erhalten, vgl. Abb. 3.5. Das ist aber nicht erwünscht, da es sich bei Core-Bereichen um besonders sichere Regionen für genau eine Klasse handeln sollte. Als wichtiger Nebeneffekt kann der Algorithmus nun *identische* Punkte *verschiedener* Klassen (durch das Zurücksetzen der Seitenausdehnungen) verarbeiten, was im Originalalgorithmus nicht möglich gewesen ist. Der nun isoliert liegende zweite Punkt erhält in der nächsten Epoche wieder die Chance zur Regelbildung.

4. Im Alg. (3.1.1), Schritt 6. wird nur geschrumpft, wenn die Regel  $p_j^l$  tatsächlich  $x$  überdeckt, d.h. wenn  $x$  in die Support-Region eines Konfliktneurons fällt; damit wird ein „unnützes“ Schrumpfen der Regeln vermieden.
5. Der optionale nearby-Parameter verhindert unrealistisch große Regeln mit großen Core-Bereichen während des Cover-Mechanismus, die dann wieder geschrumpft werden müssen. Dieser Parameter setzt allerdings eine grundsätzliche Kenntnis der Daten voraus, z.B. kann er bei Spiraldaten kleiner als der maximal mögliche Abstand  $\|(0, 0)^t - (1, 1)^t\|_2$  gesetzt werden, da im Voraus bekannt ist, dass eine Regel, die den gesamten Datenraum einnimmt, nicht existiert. Siehe hierzu die genaue Definition und die Experimente zur Voreinstellung des nearby-Parameters im Abschnitt 3.2.4.

<sup>2</sup>Die Kanten und die Punkte in den Abbildungen sind leider in schwarz-weiß nicht gut zu erkennen; es wird gebeten, besonders auf die Erläuterung zu achten.



**Abbildung 3.5:** Fuzzy-RecBF-DDA (Original): Unerwünschte Überlappung von Core-Bereichen zweier verschiedener Klassen im Originalalgorithmus.

6. Im Alg. 3.1.1, Schritt 3, wird für *alle* Prototypen  $p_i^k$  der Klasse  $k$ , die das Muster überdecken das Gewicht  $w_i^k$  um eins erhöht. Im Original wurde das Gewicht nur für *einen* Prototypen erhöht. Da die Auswahl *einer* Regel zur Erhöhung des Gewichts, z.B. diejenige mit dem größten Volumen, willkürlich erscheint und Vorabexperimente keinen Vorteil einer solchen Auswahl ergaben, werden alle entsprechenden Prototypen involviert. Da allerdings keine Performanzänderung zu erkennen ist, ist diese Änderung eher der Vollständigkeit halber erwähnt.

### 3.1.3 Bemerkung: (Klassifikation)

Ausführungsphase (asymmetrische Radien):<sup>3</sup>

$$1. \text{ Schicht: } S_i^k(x) := \min_{1 \leq d \leq n} A(x_d, z_{d,i}^k, \Lambda_{d,i,+}^k) \text{ mit} \quad (3.1)$$

$$A(x_d, z_{d,i}^k, \Lambda_{d,i,+}^k) := \begin{cases} 1 & , 0 \leq x_d - z_{d,i}^k \leq \Lambda_{d,i,+}^k \\ 0 & , \text{sonst} \end{cases} \quad (3.2)$$

und

$$T_i^k(x) := \min_{1 \leq d \leq n} A(x_d, z_{d,i}^k, \Lambda_{d,i,-}^k) \text{ mit} \quad (3.3)$$

$$A(x_d, z_{d,i}^k, \Lambda_{d,i,-}^k) := \begin{cases} 1 & , 0 \leq z_{d,i}^k - x_d \leq \Lambda_{d,i,-}^k \\ 0 & , \text{sonst} \end{cases} \quad (3.4)$$

ergeben:

$$R_i^k(x) := \max\{S_i^k(x), T_i^k(x)\} . \quad (3.5)$$

<sup>3</sup> $d$  sei eine von  $n$  Dimensionen des Datenraums.

2. Schicht: gewichtete Summe analog zum RBF-DDA (2.6):

$$y_k(x) := y_k(R_i^k(x)) := \sum_{i=1}^{m_k} w_i^k R_i^k(x) .$$

In der Ausgabe kann ein WTA-Mechanismus zur Entscheidung verwendet werden, welche Klasse gewinnt. Sind die Aktivierungen gleich, so findet keine Zuordnung zu einer Klasse statt.

Möchte man die Sensitivität und die Spezifität verändern, z.B. zur Erstellung einer ROC-Kurve, so kann man die Klassifikationsschwelle nach einer prozentualen Normierung der Ausgabeaktivierungen verändern, siehe den Abschnitt 7.5.1.

Für die Gütebewertung wurde eine *Benchmark-Umgebung* geschrieben:

#### 3.1.4 Algorithmus: (Benchmark-Modul)

- Initialisiere Parameter;
- Datenbehandlung (Normalisierung auf  $[0, 1]$  pro Variable, zufallsabhängiges Aufteilen in Trainings- und Testdaten, fehlende Werte-Behandlung);
- **while** (aktuelle Epochenzahl < max. Epochenzahl) **and**  
 ((Klassifikationsleistung < vorgegebene Klassifikationsleistung *errorcriteria*) **or**  
 aktuelle Epochenzahl < min. Epochenzahl) **do**
  - erhöhe Epochenzahl um 1;
  - permutiere Trainingsdaten;
  - führe für alle Samples des Trainingsdatensatzes den Lernalgorithmus durch;
  - Berechne Ausgabe auf den Testdaten mit Fuzzy-RecBF-DDA;
  - Berechne Klassifikationsleistung auf den Testdaten  
 (und optional auch auf den Trainingsdaten);
- end**
  - Führe Häufigkeits- und Konfidenz-Untersuchungen durch;
  - Mache Normalisierung der Daten rückgängig und gebe Regeln aus;
  - optional: Ausreißer-Behandlung (und erneutes Training, [Ber99]);
  - optional: bei 2-dimensionalen Daten Visualisierung der Regeln;
  - Speicherung der Ergebnisse und Parameter;
  - Wiederholung des Verfahrens, z.B. zehnmal, mit anschließender Berechnung der Mittelwerte, der Standardabweichungen, der Minima und der Maxima der Ergebnisse, um statistisch fundierte Aussagen zu erhalten;

Es folgt eine Auflistung der grundsätzlichen Nachteile und Vorteile des (Original-) FuzzyRec-BF-DDA-Algorithmus.

**Nachteile des Algorithmus:**

- Es kann ein leichtes overfitting (Ausrichtung auf Trainingsdaten) bzw. ein leichtes underfitting (nicht ausreichende Adaption an die Trainingsdaten) auftreten, dass aber i.d.R. erkannt wird (hohe Abweichung zwischen Trainings- und Testergebnis). Eine minimale Epochenzahl von 3 und eine maximale von 7 stellte sich als sinnvoll heraus; sinnvoll ist auch ein zusätzlicher Probedurchlauf, um zu sehen, welche vorgegebene Klassifikationsleistung man zum Abbruch mitverwenden kann. Daher ist die oben verwendete Bedingung „Klassifikationsleistung  $<$  vorgegebene Klassifikationsleistung *errorcriteria*“ in der Tat sinnvoller als eine Bedingung der Art „|aktuelle Klassifikationsleistung – Klassifikationsleistung der vorhergehenden Epoche|  $<$  Änderungsschwelle“. Diese Differenzen können sehr groß sein, so dass von einem underfitting sofort zu einem overfitting in einer weiteren Epoche übergegangen werden könnte.
- Abhängigkeit von der Reihenfolge der Präsentation der Trainingsdaten; diese Abhängigkeit im Zusammenhang mit der heuristischen Lernweise führt, insbesondere bei hoher Dimension, zu großen Core-Bereichen bei einigen Regeln, die sich im Falle vieler Überlappungen ungünstig auswirken. Dieser Aspekt wurde etwas durch die Verwendung des Schrumpfens der Core-Regeln im Vergleich zum Original-Algorithmus verbessert (Schritt 3. im Alg. 3.1.1).
- Sofortige Erzeugung einer neuen Regel bei Ausreißern (auch durch die Abhängigkeit der Präsentationsreihenfolge bedingt), die unnützlich sind (geringe Häufigkeit auf den Testdaten), siehe aber hierzu auch den Abschnitt 3.6.
- Keine direkte Verarbeitung symbolischer Daten möglich (Verwendung von Assoziations oder Generalisierungsregeln sinnvoll), siehe aber auch den nächsten Punkt.
- Unangenehme Interpretation von Variablen mit sehr wenigen diskreten Ausprägungen, z.B. binäre Variablen. Eine Ausgabe der Form „var 1 in (-infinite, 1.00)“ muss als „var 1 = 0“ oder „var 1 = 1“ interpretiert werden, da der Rand der rechteckigen Regel zur Regel dazugehört. – Gibt man mehrdimensionale binäre Daten der Form „0 x x x entspricht Klasse 1“ ( $x = 0$  oder  $1$ ) und „1 x x x entspricht Klasse 2“ vor, so findet das Verfahren unter Umständen keine Regeln aufgrund der o.g. Interpretationsvorschrift. - Generalisierungsregelverfahren finden diese beiden Regeln auf jeden Fall. Für das hier genannte Verfahren ist es also günstiger, wenn viele paarweise verschiedene Werte vorliegen. So ist es hier sinnvoll, die binären Daten zu verrauschen (z.B. 0 wird zu Werten  $0, \dots, 0.1$ ; 1 wird zu Werten  $0.9, \dots, 1$ ), da dann nicht alle 0-Werte und 1-Werte in eine Regel zusammenfallen. Dieser Vorschlag wird auch in [Wit00, S. 217-218] unterbreitet.

$d$	$u = 4^d - 1$	$v = 2d$
1	3	2
2	15	4
3	63	6
4	255	8
5	1023	10
10	1048575	20
20	$\approx 1.1 \cdot 10^{12}$	40
50	$\approx 1.3 \cdot 10^{30}$	100

**Tabelle 3.1:** Anzahl  $u$  der generell möglichen Kombinationen bei einer Regelgenerierung mit Rechtecken. Anzahl  $v$  der von Fuzzy-RecBF-DDA überprüften Möglichkeiten.

- Hohe Laufzeit bei stark überlappenden Daten mit hoher Dimensionalität, da die Laufzeit des Algorithmus von der Anzahl der erzeugten Regeln und der Dimensionalität abhängig sind. (Das ist aber eher ein allgemeingültiger Nachteil der Regelgenerierungsverfahren.)

#### Vorteile des Algorithmus:

- Verbindung von Klassifikation und Regelgenerierung.
- Heuristische Adaption mit grundsätzlicher Verringerung der Verarbeitungskomplexität (kombinatorische Explosion): Angenommen man möchte Regeln für einen  $d$ -dimensionalen Datensatz,  $d \in \mathbb{N}$ , kombinatorisch erzeugen, so entstehen bei einer Regel bereits  $2d$  Richtungen, wenn man die  $\pm$ -Richtungen betrachtet.<sup>4</sup> Jede der  $2d$  Möglichkeiten kann in Kombination auftreten, d.h. man erhält unter Verwendung des Binomischen Lehrsatzes

$$\sum_{i=1}^{2d} \binom{2d}{i} = 2^{2d} - 1 = 4^d - 1 \quad (3.6)$$

Möglichkeiten, eine Ausdehnung einer  $d$ -dimensionalen Regel vorzunehmen. Das führt bereits bei kleinerem  $d$  zu einer sehr hohen Zahl an möglichen Kombinationen, siehe die Tab. 3.1. Der Fuzzy-RecBF-DDA-Algorithmus überprüft nur  $2d$  Ausdehnungen (jede Dimension, jede Richtung). Trotz der Einsparung an überprüften Möglichkeiten, generiert der Algorithmus als adaptives System gute, suboptimale Ergebnisse.

---

<sup>4</sup>Zudem müsste man auswählen, welche der bereits vorhandenen Regeln man adaptiert bzw. ob man neue einfügt und „wieviel“ adaptiert wird.

- Erkennung redundanter Variablen innerhalb einer Regel, d.h. hat ein Teil einer Regel die Form „... **and if**  $\text{var}_j$  **in**  $(-\infty, +\infty)$  **and** ... **then** Klasse  $c$ “, dann ist die Variable  $j$  redundant für diese Regel.
- Keine explizite Vorgabe von Fuzzy-Variablen wie z.B. „niedrig“, „mittel“, „hoch“ notwendig, wenngleich möglich. Die Trapezoide adaptieren sich an die Daten, und deren Anzahl ist nicht von vornherein festgelegt.
- Einsatz des Modells für eine Funktionsapproximation ist möglich (siehe [HB96], [Hub98]).

Da der Algorithmus nur jeweils in einer Richtung einer Dimension eine Adaption vornimmt, wird die kombinatorische Explosion vermieden. Die Laufzeitordnung wird durch die Anzahl  $n$  der Samples und die Anzahl  $r$  der erzeugten Regeln dominiert. Hinzu kommt die multiplikative Abhängigkeit von der Dimensionszahl. I. Allg. ist  $r \ll n$  und  $d \ll n$ , und die mittlere Ordnung der Laufzeit einer Epoche ist  $\mathcal{O}(n \cdot r \cdot d)$ .

## 3.2 Experimente auf Benchmark-Daten

In den folgenden Abschnitten werden Experimente auf Benchmark-Daten (Datenbeschreibung siehe Anhang A.1) dokumentiert, die die Möglichkeiten des **modifizierten** Fuzzy-RecBF-DDA belegen und somit eine Anwendung auf den Realweltdaten des septischen Schocks rechtfertigen. Vor der Anwendung eines Verfahrens auf einen größeren (medizinischen) Datensatz ist es sinnvoll, das Verfahren auf kleineren, ähnlich strukturierten Datensätzen auszuprobieren, um mit den Schwächen und Stärken eines Verfahrens vertraut zu werden.

Um es noch einmal zu betonen: Es wird im Folgenden immer das von mir modifizierte Fuzzy-RecBF-DDA verwendet, da diese verbesserte Version auf allgemeinen Realweltdatensätzen durch die im Abschnitt 3.1 genannten Änderungen (z.B. Vermeidung von Core-Konflikten) robuster ist. Die durchgeführten Experimente werden immer mit dem *modifizierten Fuzzy-RecBF-DDA* durchgeführt, auch wenn der Kürze halber nur „Fuzzy-RecBF-DDA“ geschrieben steht.

Als Parameter wurde bis auf die angegebenen Abweichungen gesetzt:  $nearby = \|(0, \dots, 0)^t - (1, \dots, 1)^t\|_2 + 1$  (maximal möglicher Abstand je nach Dimension), d.h. der nearby-Parameter hat keinen Einfluss, nur bei den Spiraldaten wird er kleiner gewählt;  $\text{sigmamin} = (0.1, \dots, 0.1)$ , 50% Trainingsdaten und 50% Testdaten (bei jeder Wiederholung neu zufällig partitioniert, disjunkte Mengen), Anzahl der Wiederholungen der Versuche = 10 (Spiralen 8). Die Ergebnisse sind auf zwei Nachkommastellen angegeben, die der Leser aber aufgrund der Genauigkeit der Verfahren nicht berücksichtigen muss; die Toleranz beträgt sicher  $\pm 1\%$ .

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	96.00	3.14	88.00	100.00
Training nicht klassifiziert	0.67	1.30	0.00	4.00
Test korrekt	94.67	2.51	90.67	98.67
Test nicht klassifiziert	0.13	0.42	0.00	1.33

**Tabelle 3.2:** IRIS-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse 1	Klasse 2	Klasse 3
Klasse 1 (Training)	33.20	0.00	0.00
Klasse 2 (Training)	0.00	30.40	1.33
Klasse 3 (Training)	0.00	2.00	32.40
Klasse 1 (Test)	33.33	0.00	0.00
Klasse 2 (Test)	0.13	31.20	2.40
Klasse 3 (Test)	0.00	2.67	30.13

**Tabelle 3.3:** IRIS-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

### 3.2.1 Experiment mit den IRIS-Daten

Die IRIS-Daten dienen zum Testen der Funktionstüchtigkeit des Verfahrens auf mehr als 2 Klassen. Die IRIS-Daten sind als leicht zu klassifizierendes Problem bekannt. Das gute Klassifikationsergebnis und die geringe Regelzahl belegen diesen Sachverhalt.

Sowohl in diesem als auch in den anderen experimentellen Abschnitten werden in Kürze die Versuchseinstellungen beschrieben und die Ergebnisse der Klassifikation und der Regelgenerierung präsentiert. Im Abschnitt 3.2.5 werden die Ergebnisse auf den verschiedenen Testdatensätzen zusammenfassend bewertet.

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 90\%$  (minimale Klassifikationsleistung, bei der nach frühestens 3 Epochen das Lernen abgebrochen wird, um unnötige Epochen zu vermeiden; der Parameter kann durch Vorversuche empirisch bestimmt werden), Ergebnisse siehe Tabellen 3.2 und 3.3 (Klassifikation) und Tabellen 3.4, C.1 und C.2 im Anhang C (Regelgenerierung). Die verwendeten Testdaten kamen in den 10 Testdurchläufen im Mittel zu 33.47% aus Klasse 1, zu 33.87% aus Klasse 2 und zu 32.67% aus Klasse 3. Die Ergebnisse sind immer in Relation zu der Klassenverteilung zu sehen. Die Epochenzahl betrug im Mittel 3.0 mit einer Standardabweichung von 0 (Minimum 3, Maximum 3).

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1	1.0	0.0	1	1
Klasse 2	2.5	0.9	1	3
Klasse 3	2.5	0.9	1	4

**Tabelle 3.4:** IRIS-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

Auf dieser Seite werden die Regeln des neunten der zehn Durchläufe wiedergegeben.

**Wichtig:** Beträgt die Häufigkeit einer Regel 0, so wird auch die Konfidenz konservativ auf 0 gesetzt (auch bei allen folgenden Experimenten). Die Gütemaße, die hier immer auf den Testdaten berechnet wurden, sind  $\cdot 100$  als „%“ zu verstehen. Die runden Klammern bedeuten „ $\geq$ “ bzw. „ $\leq$ “. Die jeweils irrelevanten Dimensionen werden nicht aufgeführt. Zur Angabe der Regeln vgl. auch den Abschnitt 3.4.

**1)** support class **1** with test frequency **0.387** and test confidence **1.000** if  
**var 3**  $\leq 3.00$

core class **1** with test frequency **0.307** and test confidence **1.000** if

**var 1** in (4.30,5.50) and

**var 2** in (2.90,4.20) and

**var 3** in (1.10,1.90) and

**var 4** in (0.10,0.50)

**2)** support class **2** with test frequency **0.347** and test confidence **0.846** if

**var 3**  $\geq 1.70$  and

**var 4**  $\leq 1.80$

core class **2** with test frequency **0.240** and test confidence **1.000** if

**var 1** in (4.90,6.90) and

**var 2** in (2.20,3.40) and

**var 3** in (3.00,4.90) and

**var 4** in (1.00,1.60)

**3)** support class **2** with test frequency **0.133** and test confidence **0.500** if

**var 1**  $\leq 6.40$  and

**var 2**  $\geq 2.80$  and

**var 4** in (0.50,2.30)

core class **2** with test frequency **0.053** and test confidence **1.000** if

**var 1** in (5.40,6.10) and

**var 2** in (2.80,3.20) and

**var 3** in (4.10,4.80) and

**var 4** in (1.20,1.80)

4) support class **3** with test frequency **0.267** and test confidence **1.000** if  
**var 4**  $\geq 1.80$   
core class **3** with test frequency **0.200** and test confidence **1.000** if  
**var 1** in (5.80,7.90) and  
**var 2** in (2.50,3.80) and  
**var 3** in (5.10,6.40) and  
**var 4** in (1.80,2.50)

5) support class **3** with test frequency **0.320** and test confidence **0.917** if  
**var 1**  $\geq 5.60$  and  
**var 3**  $\geq 4.80$   
core class **3** with test frequency **0.267** and test confidence **0.900** if  
**var 1** in (5.80,7.90) and  
**var 2** in (2.50,3.80) and  
**var 3** in (4.90,6.90) and  
**var 4** in (1.60,2.50)

Man sieht, dass alle Regeln bis auf die Regel 3) eindeutig gut sind, d.h. eine hohe Häufigkeit und Konfidenz besitzen. Die Regel 3) beschreibt eine Überlappungsregion von Daten. Für diesen einfach strukturierten Datensatz erhält man ein sehr gutes Klassifikationsergebnis und Regeln, die eine sinnvolle Erklärung der Klassifikationsentscheidung liefern.

### 3.2.2 Experiment mit den Cancer1-Daten

Auch der Cancer1-Datensatz ist ein einfach zu klassifizierender medizinischer Datensatz. Bei diesem Zwei-Klassen-Problem wird zusätzlich zur Beurteilung der Klassifikationsleistung ein ROC-Kurven-Flächeninhalt berechnet (wie auch bei anderen Experimenten mit zwei Klassen).

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 90$ , Ergebnisse siehe Tabellen 3.5 und 3.6 (Klassifikation) und Tabellen 3.7, C.3 und C.4 im Anhang C (Regelgenerierung). Die verwendeten Testdaten kamen in den 10 Testdurchläufen im Mittel zu 35.14% aus Klasse 1 und zu 64.86% aus Klasse 2. Die Epochenzahl betrug im Mittel 3.1 mit einer Standardabweichung von 0.32 (Minimum 3, Maximum 4).

In der Anwendungsphase erhält man für jeden Eingabevektor eine Ausgabeaktivierung pro Klasse, die man prozentual normieren kann (siehe den Abschnitt 7.5.1). Verändert man die Klassifikationsschwelle in 10%-Schritten, ab der ein Eingabevektor als Klasse 2 klassifiziert wird, so kann man 10 Paare (Spezifität, Sensitivität) berechnen, die eine ROC-Kurve bilden. Die Schwelle steuert das Sensitivitäts-

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	96.10	0.49	94.84	96.56
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	95.94	0.88	94.57	97.14
Test nicht klassifiziert	0.06	0.12	0.00	0.29

**Tabelle 3.5:** Cancer1-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse 1	Klasse 2
Klasse 1 (Training)	32.32	2.41
Klasse 2 (Training)	1.49	63.78
Klasse 1 (Test)	33.20	2.06
Klasse 2 (Test)	1.94	62.74

**Tabelle 3.6:** Cancer1-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

Spezifitäts-Verhalten des Klassifikators. Ein einzelnes Paar auf der ROC-Kurve ist nur bedingt aussagekräftig.

In der Abb. B.1 sind alle Klassifikationsergebnisse der 10 Versuche aufgetragen, wobei jeweils 10 Paare der o.g. Art eingetragen wurden, also insgesamt 100 Klassifikationen durchgeführt wurden. Berechnet man aus den einzelnen 10 ROC-Kurven den Flächeninhalt unter den Kurven, so kann man dann den mittleren Flächeninhalt unter den 10 Kurven berechnen; der Flächeninhalt beträgt jeweils 0.5, falls die ROC-Kurve gerade die Diagonale darstellt und nähert sich dem Wert 1, falls die Kurve sich nach oben rechts „ausdehnt“. Für die Cancer1-Daten erhält man den sehr guten Wert 0.987 (Min. 0.982, Max. 0.996).

Bei den erzeugten Regeln für den Cancer1-Datensatz fällt auf, dass einige Core-Regeln nicht mehr von den Testdaten getroffen werden, d.h. sie haben eine Core-Häufigkeit von 0. Auch gibt es neben einigen „guten“ Regeln auch einige „schlechte“.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1	4.9	0.7	4	6
Klasse 2	5.6	1.4	4	7

**Tabelle 3.7:** Cancer1-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

Diese „schlechten“ Regeln sind wahrscheinlich durch Ausreißer-Daten entstanden; diese Regeln beeinträchtigen zwar nicht so sehr die Klassifikationsleistung, tragen aber zur unnötigen Verkleinerung anderer Regeln bei ([BH98] und Abschnitt 3.3). Exemplarisch sei je eine „gute“ und eine „schlechte“ Regel pro Klasse wiedergegeben (Regeln 2, 4, 6 und 11 aus dem 2. von 10 Testdurchläufen mit der entsprechenden dort verwendeten Nummerierung):

2) support class **1** with test frequency **0.449** and test confidence **0.758** if

**var 6**  $\geq 0.10$  and

**var 8**  $\geq 0.10$

core class **1** with test frequency **0.263** and test confidence **0.946** if

**var 1** in (0.50,1.00) and

**var 2** in (0.30,1.00) and

**var 3** in (0.30,1.00) and

**var 4** in (0.10,1.00) and

**var 5** in (0.20,1.00) and

**var 6** in (0.30,1.00) and

**var 7** in (0.30,0.90) and

**var 8** in (0.10,1.00) and

**var 9** in (0.10,1.00)

4) support class **1** with test frequency **0.100** and test confidence **0.257** if

**var 1**  $\geq 0.10$  and

**var 3**  $\geq 0.10$  and

**var 4**  $\leq 0.20$  and

**var 8**  $\leq 0.40$

core class **1** with test frequency **0.000** and test confidence **0.000** if

**var 1** in (0.70,1.00) and

**var 2** in (0.20,0.40) and

**var 3** in (0.40,0.60) and

**var 4** = 0.10 and

**var 5** in (0.20,0.30) and

**var 6** in (0.40,1.00) and

**var 7** in (0.30,0.50) and

**var 8** = 0.30 and

**var 9** = 0.10

6) support class **2** with test frequency **0.577** and test confidence **0.970** if

**var 1**  $\leq 0.50$  and

**var 4**  $\leq 0.40$  and

**var 8**  $\leq 0.30$

core class **2** with test frequency **0.566** and test confidence **0.990** if  
**var 1** in (0.10,0.50) and  
**var 2** in (0.10,0.40) and  
**var 3** in (0.10,0.60) and  
**var 4** in (0.10,0.30) and  
**var 5** in (0.10,0.50) and  
**var 6** in (0.10,0.50) and  
**var 7** in (0.10,0.70) and  
**var 8** in (0.10,0.30) and  
**var 9** in (0.10,0.80)

11) support class **2** with test frequency **0.060** and test confidence **0.381** if  
**var 2**  $\leq 1.00$  and  
**var 3**  $\leq 0.50$  and  
**var 4**  $\geq 0.40$  and  
**var 8**  $\geq 0.10$

core class **2** with test frequency **0.014** and test confidence **0.600** if  
**var 1** in (0.10,0.80) and  
**var 2** in (0.10,0.40) and  
**var 3** in (0.10,0.40) and  
**var 4** in (0.40,0.50) and  
**var 5** in (0.10,0.40) and  
**var 6** in (0.10,0.70) and  
**var 7** in (0.10,0.70) and  
**var 8** in (0.10,0.80) and  
**var 9** in (0.10,0.20)

Schön zu sehen ist bei den Regeln, dass im Support-Teil nur wenige der neun Variablen benötigt wurden und einige redundant waren. Daraus resultiert eine bessere Verständlichkeit der Support-Regeln im Vergleich zu den Core-Regeln, die immer alle 9 Dimensionen berücksichtigen.

### 3.2.3 Experiment mit den Diabetes1-Daten

Die Diabetes1-Daten zeigen ein stärkeres Überlappungsverhalten und können am ehesten als Benchmark für eine Anwendung auf umfangreicheren medizinischen Daten dienen.

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 70$ , Ergebnisse siehe Tabellen 3.8 und 3.9 (Klassifikation) und Tabellen 3.10, C.5 und C.6 im Anhang C (Regelgenerierung). Die verwendeten Testdaten kamen den 10 Testdurchläufen im Mittel zu 64.56% aus Klasse 1 und zu 35.44% aus Klasse 2. Die Epochenanzahl betrug im Mittel 4.3 mit einer Standardabweichung von 1.42 (Minimum 3, Maximum 7).

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	73.31	1.84	69.79	75.00
Training nicht klassifiziert	0.08	0.18	0.00	0.52
Test korrekt	70.89	1.58	66.93	72.66
Test nicht klassifiziert	0.16	0.35	0.00	1.04

**Tabelle 3.8:** Diabetes1-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse 1	Klasse 2
Klasse 1 (Training)	58.78	19.77
Klasse 2 (Training)	6.85	14.53
Klasse 1 (Test)	56.77	21.33
Klasse 2 (Test)	7.63	14.12

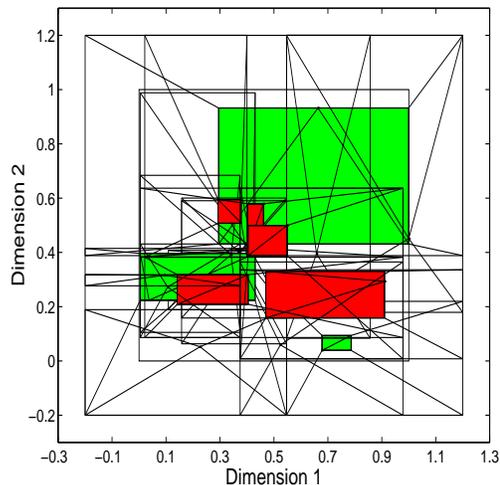
**Tabelle 3.9:** Diabetes1-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

Auf die Angabe von Regeln wird aus Platzgründen verzichtet. Auch hier seien 10 ROC-Kurven abgebildet, um das Klassifikationsergebnis besser einschätzen zu können im Hinblick auf die Veränderung der Klassifikationsschwelle (Abb. B.2). Alle Klassifikationsergebnisse der 10 Versuche sind aufgetragen, wobei jeweils 10 Paare eingetragen wurden, also insgesamt 100 Klassifikationen durchgeführt wurden. Darüber hinaus wurden die Ergebnisse der 100 Klassifikationen aus ebenfalls 10 Versuchen mit einer Vorverarbeitung der linearen ICA „FastICA“ (siehe Abschnitt 2.4.3) abgebildet.

Der mittlere Flächeninhalt unter den ROC-Kurven beträgt 0.725 (Min.: 0.656 , Max.: 0.799) ohne LICA und 0.742 (Min.: 0.706 , Max.: 0.798) mit LICA. Die Ergebnisse der Klassifikation (und auch der Regelgenerierung) sind mit oder ohne der linearen ICA nahezu identisch (z.B. auch Regelanzahl im Mittel ohne LICA: 35.9, mit LICA: 36.4), so dass auf eine genaue Angabe der Ergebnisse bez. der Diabetes1-Daten mit

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1	18.7	1.6	16	21
Klasse 2	16.2	3.6	12	21

**Tabelle 3.10:** Diabetes1-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).



**Abbildung 3.6:** Generierte Regeln aus den Spiraldaten (1. Versuch),  $nearby = 2$  (ohne).

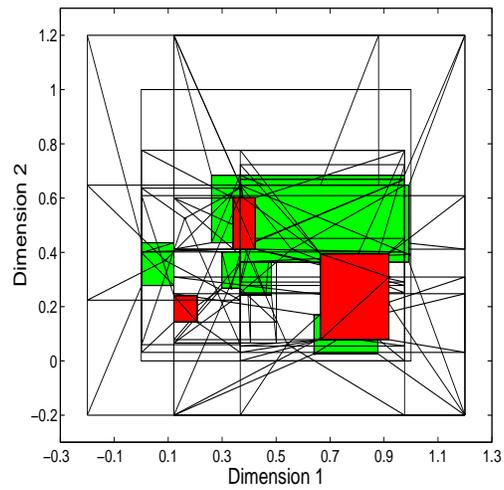
Vorverarbeitung durch eine LICA verzichtet wird. Die Hoffnung einer Steigerung der Performanz durch die Vorverarbeitung mit einer LICA hat sich hier nicht bestätigt.

### 3.2.4 Experiment mit den Spiraldaten

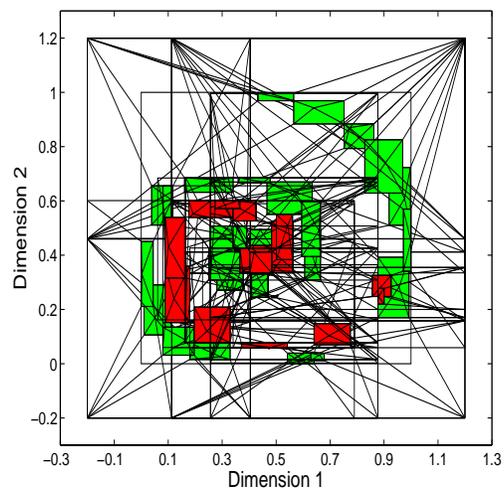
Anhand der Spiraldaten soll vor allem die Nützlichkeit des Parameters  $nearby$ , definiert in (3.7) auf S. 87, zur Einschränkung der maximalen Regelgröße gezeigt werden.

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $errorcriteria = 65$ , eine Visualisierung der Daten findet man in der Abb. A.1 im Anhang A. 1. Tabellenserie:  $nearby = 2$  (d.h. so groß gewählt, dass er keine Rolle spielt, kurz: „ohne“), 2. Tabellenserie:  $nearby = 0.14$ . Ergebnisse siehe Tabellen 3.11 und 3.12 bzw. 3.15 und 3.16 (Klassifikation) und Tabellen 3.13, C.7 und C.8 bzw. 3.17, C.9 und C.10 (Regelgenerierung). Die verwendeten Testdaten kamen in den 8 Testdurchläufen in der 1. Tabellenserie im Mittel zu 50.13% aus Klasse 1 und zu 49.87% aus Klasse 2. Die Epochenzahl betrug im Mittel 4.1 mit einer Standardabweichung von 1.8 (Minimum 3, Maximum 7). In der 2. Tabellenserie kamen die Daten im Mittel zu 48.87% aus Klasse 1 und zu 51.13% aus Klasse 2. Die Epochenzahl betrug im Mittel 3.6 mit einer Standardabweichung von 1.06 (Minimum 3, Maximum 6).

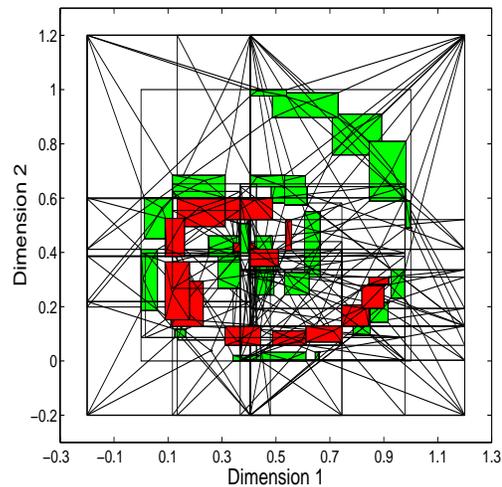
Die beiden Abbildungen 3.6 und 3.7 (bzw. 3.8 und 3.9) erzeugter Regeln auf den Spiraldaten zeigen deutlich die Abhängigkeit des Verfahrens von der Reihenfolge der Datenpräsentation bzw. von der Auswahl der Testdaten. Im Folgenden werden die Ergebnisse präsentiert, einmal mit  $nearby = 2$  (ohne) und einmal mit  $nearby = 0.14$  (und  $a = 1$ ). Die Abbildungen zeigen auch deutlich die Grenzen des Verfahrens, wenn man wie in [HB95] ohne  $nearby$ -Parameter arbeitet.



**Abbildung 3.7:** Generierte Regeln aus den Spiraldaten (2. Versuch),  $nearby = 2$  (ohne).



**Abbildung 3.8:** Generierte Regeln aus den Spiraldaten (1. Versuch),  $nearby = 0.14$ .



**Abbildung 3.9:** Generierte Regeln aus den Spiraldaten (2. Versuch),  $nearby = 0.14$ .

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	66.44	3.44	61.32	69.55
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	64.84	4.35	57.41	68.72
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 3.11:** Spiraldaten ( $nearby = 2$  (ohne)): korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

Dabei berechnet sich der nearby-Parameter folgendermaßen: Seien  $(x_i; k_i)$  die Muster aus einem Datensatz  $\mathcal{DS}$  mit Klassenzugehörigkeit  $k_i$ .  $d_{ij}$  seien die Abstände zwischen den Punkten unterschiedlicher Klassenzugehörigkeit, d.h.  $d_{ij} := \|(x_i; k_i) - (x_j; k_j)\|_2$ .  $m$  sei die Anzahl der Abstände. Die prinzipielle Idee zur Voreinstellung eines Abstandes, bei dessen Überschreitung anstelle eines Cover-Schritts eine neue Regel erzeugt wird (nearby-Parameter), ist die Berücksichtigung der Abstände von Punkten verschiedener Klassen. Interessant sind der mittlere und der minimale Abstand. Da der mittlere Abstand alle Daten repräsentiert, aber der minimale Abstand die kritische untere Grenze bei der Regelerzeugung darstellt, wird folgendes Maß für den nearby-Parameter gewählt, so dass durch  $a \in [0, 1]$  der Einfluss des minimalen Abstandes variiert werden kann. Dabei sei  $c$  die Anzahl der Klassen,  $K_k$  bzw.  $L_l$  die Anzahl der Regeln einer Klasse  $k$  bzw.  $l$ . Das Minimum wird über alle berechneten

	Klasse 1	Klasse 2
Klasse 1 (Training)	24.28	8.08
Klasse 2 (Training)	25.49	42.05
Klasse 1 (Test)	23.89	8.92
Klasse 2 (Test)	26.24	40.95

**Tabelle 3.12:** Spiraldaten ( $nearby = 2$  (ohne)): Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1	7.6	1.2	6	9
Klasse 2	10.3	1.8	8	14

**Tabelle 3.13:** Spiraldaten ( $nearby = 2$  (ohne)): Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

Abstände  $d_{ij}$  gebildet ( $1 \leq i \leq K_k$ ,  $1 \leq j \leq L_l$ ,  $k = 1, \dots, c$ ,  $l = k + 1, \dots, c$ ).

$$nearby := \left( \frac{1}{m} \sum_{k=1}^c \sum_{l=k+1}^c \sum_i^{K_k} \sum_j^{L_l} d_{ij} + a \cdot \min\{d_{ij}\} \right) / (1 + a) . \quad (3.7)$$

In der Tab. 3.14 sind einige Werte für den nearby-Parameter angegeben inkl. den Mittelwerten, Standardabweichungen, Minima und Maxima. Als Fazit kann man sagen, dass die nearby-Parameter der Datensätze IRIS und Cancer1 auf potenziell gut trennbare Klassen hinweisen; die Klassen der Datensätze Diabetes1, Spiral und UniformNoise sind eher schlecht trennbar, was auch den den Tatsachen entspricht.

### 3.2.5 Bewertung der Ergebnisse

In [Pre94] wurde ein Klassifikationsergebnis von 98.85% (Cancer1) bzw. von 75.00% (Diabetes1) auf den Testdaten eines mit einem Backpropagation trainierten FF-Netzes festgestellt. Die mit Fuzzy-RecBF-DDA erreichten Ergebnisse (95.94% bzw. 70.89%, siehe Tabelle 3.5 bzw. 3.8) erreichen diese Klassifikationsleistung nicht ganz. Allerdings hat man den Vorteil der Regelgewinnung. Die IRIS-Daten können i.d.R. von allen Verfahren gut klassifiziert werden; Unterschiede können dort nicht als signifikant gewertet werden. Die vorgeschlagene Einführung eines nearby-Parameters beeinflusst das Ergebnis positiv bei den Spiraldaten. Der Vorteil hierbei ist eine deutlich bessere Konfidenz bei allerdings höherer Regelanzahl mit niedrigerer Häufigkeit.

	Mittelwert	Standardabw.	Minimum	Maximum	nearby
<i>IRIS</i>					
absolut	0.82	0.32	0.09	1.65	0.46
prozentual	40.98	15.79	4.39	82.26	22.69
<i>Cancer1</i>					
absolut	1.72	0.45	0.29	2.86	1.01
prozentual	57.23	14.85	9.80	95.37	33.52
<i>Diabetes1</i>					
absolut	0.65	0.21	0.05	1.70	0.35
prozentual	22.81	7.46	1.87	59.97	12.34
<i>Spiral</i>					
absolut	0.28	0.20	0.01	0.95	0.15
prozentual	20.08	13.89	0.55	67.14	10.32
<i>UniformNoise</i>					
absolut	0.51	0.24	0.00	1.33	0.26
prozentual	36.07	17.11	0.01	94.27	18.04

**Tabelle 3.14:** Nearby-Parameter verschiedener Datensätze (Mittelwert, Standardabweichung, Minimum, Maximum der Abstände, nearby-Parameter, jeweils absolut im  $[0, 1]^{\dim \mathcal{D}^S}$ ) und relativ in Prozent zum maximal möglichen Abstand (Die absoluten Abstände sind dimensionsabhängig.),  $a = 1$ . „UniformNoise“ ist ein 2-dim. gleichverteiltes Rauschen mit 700 Datenpunkten.

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	69.91	6.31	62.14	82.30
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	68.36	3.92	65.02	75.51
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 3.15:** Spiraldaten ( $nearby = 0.14$ ): Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse 1	Klasse 2
Klasse 1 (Training)	28.99	7.95
Klasse 2 (Training)	22.15	40.92
Klasse 1 (Test)	26.62	9.39
Klasse 2 (Test)	22.25	41.74

**Tabelle 3.16:** Spiraldaten ( $nearby = 0.14$ ): Fehler auf den Trainings- und Testdaten (Zeile: Klassenzugehörigkeit, Spalte: Ergebnis der Klassifikation). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1	26.6	2.0	24	29
Klasse 2	17.1	2.4	15	22

**Tabelle 3.17:** Spiraldaten ( $nearby = 0.14$ ): Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

Neben den positiven Aspekten fällt negativ auf, dass ein Teil der Testdaten nicht klassifiziert werden kann; das kann aber prinzipiell bei allen Verfahren passieren, die einen WTA-Mechanismus vorsehen oder eine minimale Aktivierungsdifferenz (zur Sicherheit des Ergebnisses) voraussetzen. Der Anteil der nicht klassifizierbaren Daten ist i. Allg. aber sehr klein, so dass der Nachteil nicht bedeutend ist.

Schwerwiegender für die Regelgenerierung ist die Tatsache, dass das Verfahren, insbesondere bei hoher Dimension, neben „guten“ Regeln auch „schlechte“ generiert, was z.B. bei den Spiraldaten zu schlechteren Durchschnittsergebnissen führt. Wie bereits gesagt, kann das nicht vollständig den Daten zugeschrieben werden. Es handelt sich auch um eine Schwäche des Verfahrens, da für „isolierte“ Punkte „kleine“ Regeln erzeugt werden. Die vorgeschlagene Einführung eines nearby-Parameters beeinflusst das Ergebnis positiv (Spiraldatenversuch). Aus der Abhängigkeit von der Reihenfolge entstehen auch die Standardabweichungen und Unterschiede zwischen Minima und Maxima bei den Werten zur Regelgenerierung (Anzahl der Regeln, Core-Konfidenz, Support-Konfidenz, Core-Häufigkeit, Support-Häufigkeit). Allerdings muss auch berücksichtigt werden, dass für eine Häufigkeit auf der Testmenge gleich Null auch die Konfidenz auf Null gesetzt wurde, was z.B. bei den Diabetes1-Daten dazu führt, dass die Konfidenz bei den Support-Regeln größer als bei den Core-Regeln ist, da die Core-Regeln oftmals nicht von den Testdaten „getroffen“ werden.

Eine weitere Verbesserungsmöglichkeit besteht in der Angabe der Ergebnisse zu einem bestimmten Fuzzy-Niveau (=Grad der Zugehörigkeit) zwischen 0 und 1 (z.B. 0.1,...,0.9 anstelle von Core = 1.0 und Support = 0.0) mit anschließender Suche nach

einer geeigneten scharfen Regel, siehe weiter unten den Abschnitt 3.4. Dadurch kann der Gegensatz zwischen Häufigkeit und Konfidenz angepasst werden.

Eine Bewertung des Regelgenerators (inkl. der Klassifikationsleistung) auf den Vorstudien- und Klinikdaten erfolgt in den Kapiteln 5 und 7.

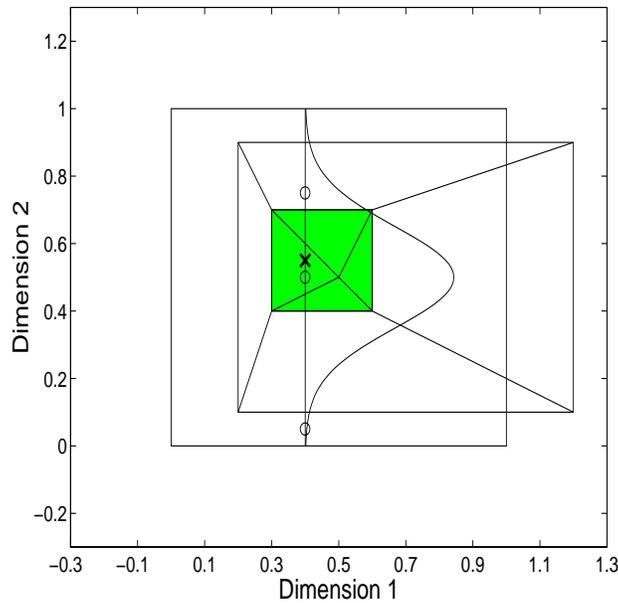
### 3.3 Behandlung fehlender Werte

In den oben genannten Benchmark-Datensätzen gibt es keine *fehlenden Werte* (engl.: Missing Values), d.h. fehlende Werte in einzelnen Dimensionen. Treten fehlende Werte auf, so gibt es verschiedene Möglichkeiten der Behandlung, die man im Wesentlichen in informative, Kontext-basierte (z.B. Mittelwerte einfügen oder lokale Regressionen) und nicht-informative (z.B. Zufallswerte einfügen) Verfahren einteilen kann [Wit00, Abschnitt 3.2].

Die informative Methode aus [BH98] wird hier nicht verwendet, da die dort vorgeschlagene, in Abb. 3.10 visualisierte *Schwerpunktmethode* die fehlenden Daten sehr „optimistisch“ anhand der bereits trainierten vollständigen Daten einordnet. Bei sehr vielen fehlenden Werten kann es, wie auch bei einer *klassenabhängigen* Mittelwertsersetzung, dazu kommen, dass das Klassifikationsergebnis mit Ersetzungswerten sogar besser ist als das Resultat auf einem entsprechenden vollständigen Datensatz, der keine fehlenden Werte besitzt (absurd!). Informative Verfahren eignen sich daher eher, wenn nur wenige fehlende Werte vorhanden sind. Die MEDAN-Daten enthalten viele fehlende Werte, so dass eine nicht-informative Strategie gewählt wird.

Für medizinische Variablen, die annähernd normalverteilt sind, bietet sich als einfache nicht-informative Methode eine zufällige Einordnung um den Mittelwert gemäß eines Normalverteilungs-Zufallsgenerators an, siehe die Abb. 3.10. Diese Methode, im Folgenden *meannoise* genannt, wird auf die Vorstudien Daten angewendet. In der Abb. 3.10 wird nach [BH98] der Punkt (0.4, mv) (mv = missing value) auf das Kreuz in den Schwerpunkt der übrigen Dimensionen, d.h. hier der 2. Dimension, gesetzt. In der nicht-informativen Variante *meannoise* wird der Punkt zufällig gemäß der eingezeichneten Normalverteilung einem Wert zugeordnet, d.h. einem der Kreise.

Da die Diabetes1-Daten am ehesten den MEDAN-Daten entsprechen in Bezug auf Überlappung und Datenqualität, werden Experimente mit fehlenden Werten auf dem Diabetes1-Datensatz durchgeführt. Die Testbedingungen werden nicht verändert mit der Ausnahme, dass 20% fehlende Werte zufällig eingefügt wurden. Das Ziel ist hier nicht zu zeigen, dass das Verfahren mit fehlenden Werten gleich gut arbeitet oder gar besser wird, sondern zu zeigen, inwieweit fehlende Werte das Ergebnis (negativ) beeinflussen. Das auf diesem Resultat erzielte Ergebnis kann nicht ohne weiteres auf andere Datensätze übertragen werden; es gibt nur einen Hinweis auf die Größenordnung des Fehlers, der entsteht.



**Abbildung 3.10:** Fuzzy-RecBF-DDA (Beispiel): Ersetzung eines fehlenden Wertes. Variante *Schwerpunkt* (Kreuz) im Vergleich zur Variante *meannoise* (einer der Kreise).

**Diabetes1 mit fehlenden Werten:** vorgegebene Klassifikationsleistung  $\text{errorcriterium} = 70$ , Ergebnisse siehe Tabellen 3.18 und 3.19 (Klassifikation) und Tabellen 3.20, C.11 und C.12 (Regelgenerierung). Die verwendeten Testdaten kamen in den 10 Testdurchläufen im Mittel zu 65.08% aus Klasse 1 und zu 34.92% aus Klasse 2. Die Epochenzahl betrug im Mittel 4.5 mit einer Standardabweichung von 1.78 (Minimum 3, Maximum 7).

Das Ergebnis der Diabetes1-Klassifikation mit fehlenden Werten (*meannoise*) entspricht innerhalb der Toleranzgrenzen in etwa dem Ergebnis der Diabetes1-Klassifi-

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	72.29	2.78	67.19	76.56
Training nicht klassifiziert	0.05	0.16	0.00	0.52
Test korrekt	71.41	3.06	66.93	75.52
Test nicht klassifiziert	0.10	0.18	0.00	0.52

**Tabelle 3.18:** Diabetes1-Daten (fehlende Werte): korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse 1	Klasse 2
Klasse 1 (Training)	56.38	18.93
Klasse 2 (Training)	8.72	15.91
Klasse 1 (Test)	56.22	19.66
Klasse 2 (Test)	8.83	15.18

**Tabelle 3.19:** Diabetes1-Daten (fehlende Werte): Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1	21.4	2.8	16	26
Klasse 2	17.6	2.6	14	22

**Tabelle 3.20:** Diabetes1-Daten (fehlende Werte): Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

kation ohne fehlende Werte. Allerdings ist die mittlere Anzahl der erzeugten Regeln pro Klasse bei fehlenden Werten etwas höher (mit fehlenden Werten Kl. 1: 21.40, Kl. 2: 17.60; ohne fehlende Werte Kl. 1: 18.70, Kl. 2: 16.20), die mittlere Konfidenz der Core-Regeln etwas niedriger (mit fehlenden Werten Kl. 1: 47.37%, Kl. 2: 27.95%; ohne fehlende Werte Kl. 1: 52.50%, Kl. 2: 31.76%) und auch die mittlere Häufigkeit der Core-Regeln etwas niedriger (mit fehlenden Werten Kl. 1: 2.71%, Kl. 2: 1.26%; ohne fehlende Werte Kl. 1: 4.37%, Kl. 2: 1.65%).

Aufgrund der Vorüberlegungen und -experimente kann davon ausgegangen werden, dass dieser Ansatz zur Ersetzung von fehlenden Werten, nicht zu einer unzulässigen Verbesserung des Klassifikationsergebnisses auf den Vorstudien-daten führt, aber dennoch die vorhandenen Werte sinnvoll ausgewertet werden können.

Es stellen sich nun aber drei Fragen:

1. Wie groß sollte die Standardabweichung der Ersetzungswerte sein?
2. Was sollte man tun, wenn die Verteilungen nicht annähernd normalverteilt sind?
3. Sollte man nicht besser die mehrdimensionale Verteilung anstelle eindimensionaler Randverteilungen betrachten?

Zu 1.: Setzt man jeden Wert auf den Mittelwert, so hat die resultierende Verteilung eine geringere Varianz. Setzt man zufällig gemäß der Verteilung ein, so könnten zufällig Ausreißer als Ersetzungswert generiert werden, was sehr unerwünscht ist.

Sinnvoll ist, Ersetzungswerte im „Kernbereich“ der Verteilung zu erzeugen, z.B. in 5 von 10 Histogrammstäben der Verteilung, was in etwa dem sog. *Interquartilsbereich* entspricht. Zu 2.: Neben einer Parameterschätzung bietet sich eine Histogrammschätzung an. Generiert man gemäß Punkt 1. Ersetzungswerte nicht aus allen Histogrammstäben, so kann man für eingipflige schiefe Verteilungen 5 Histogrammstäbe um den *Median* (anstelle des Mittelwertes) herum auswählen und dort gemäß einer Normalverteilung Ersetzungswerte generieren. Zu 3.: I.d.R. sind in der medizinischen Datenanalyse nicht genügend Daten für eine mehrdimensionale Schätzung vorhanden, abgesehen von dem größeren Aufwand, den man für einige Ersetzungswerte betreiben würde.

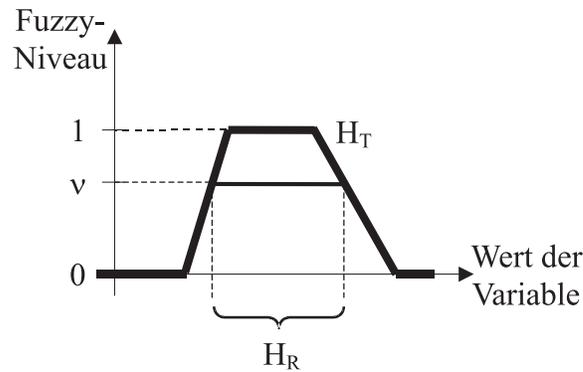
Für die Klinikdaten wird aufgrund der obigen Überlegungen nicht die Variante *meannoise* verwendet, sondern es wird die (bessere) Variante *mediannarrownoise* verwendet, d.h. ein schmales normalverteiltes Rauschen (5 von 10 Histogrammstäbe) um den Median gemäß den diskutierten Punkten 1. und 2. Der Median ist im Vergleich zum Mittelwert besser für Verteilungen geeignet, die nicht annähernd normalverteilt sind.

Ergänzung: Von einer *Ausreißer-Behandlung* gemäß [Ber99] oder gemäß Abschnitt 3.6 wird bei der Analyse der MEDAN-Daten abgesehen, da bei den MEDAN-Daten aufgrund der vielen Überlappungen nicht klar ist, wann es sich bei Datensamples um Ausreißer handelt und wann nicht. In [Ber99] werden die Regeln (inkl. der zugehörigen Daten) mit geringer Häufigkeit entfernt und es wird erneut trainiert.

### 3.4 Gewinnung scharfer Regeln durch schrittweise Erhöhung des Fuzzy-Niveaus

Bei der Bestimmung des Klassifikationsergebnisses werden die Testdaten verwendet, nicht die Trainingsdaten. Sind die Datensätze insgesamt nicht repräsentativ, d.h. treten bei einer zufälligen Einteilung der Datensätze in Trainings- und Testdaten Unterschiede in den jeweiligen Klassifikationsergebnissen der Versuchswiederholungen auf (hohe Standardabweichung des Klassifikationsergebnisses), so tritt der im Folgenden beschriebene, praktisch bedeutsame Effekt auf.

Wie wir in diesem Abschnitt feststellen werden, kann die Ausgabe von Fuzzy-Regeln problematisch sein wegen der geringen Konfidenz der Core-Regeln, da aufgrund des kleinen Volumens des Core-Bereichs der Trapezoide oft keine Testdaten im Core-Bereich liegen. Es bietet sich daher eine Überprüfung der Qualität der (weiter unten definierten) *scharfen* Regeln an, die man aus den trapezoiden Zugehörigkeitsfunktionen generieren kann.



**Abbildung 3.11:** Konstruktion scharfer Regeln: Eine 1-dimensionale Zugehörigkeitsfunktion, interpretiert als ein Trapezoid  $H_T$  und ein 1-dimensionales Hyper-Rechteck (= Intervall)  $H_R$ , herausgeschnitten aus  $H_T$  auf der Höhe des Fuzzy-Niveaus  $\nu$ .

Wie bereits erwähnt, muss man dabei die Definition (scharfer) Regeln nicht auf die Core- bzw. Support-Regeln beschränken (Fuzzy-Niveau 1.0 bzw. 0.0). Deshalb werden nun auch die Zwischenstufen 0.1, ..., 0.9 betrachtet.

Für unsere Zwecke soll genau definiert werden, was unter einer scharfen Regel verstanden werden soll, die aus einem Trapezoid entsteht. Zur Gütebewertung der entstehenden scharfen Regeln werden die Maße Häufigkeit und Konfidenz aus dem Abschnitt 2.5.2 übertragen.

### 3.4.1 Definition: (Scharfe Regel)

a) Sei  $H_T$  ein durch das Verfahren Fuzzy-RecBF-DDA erzeugtes Trapezoid. Eine **scharfe Regel**  $R$  ist definiert durch ein Hyper-Rechteck  $H_R$ , dass aus  $H_T$  auf der Höhe eines gewählten Fuzzy-Niveaus (Zugehörigkeitsgrad) herausgeschnitten wird, siehe die Abb. 3.11. Hat der Support-Bereich von  $H_T$  in einer Dimension unendliche Seitenausdehnungen, so wird er für die Gewinnung scharfer Regeln auf den maximalen bzw. minimalen (je nach Richtung) vorkommenden Wert in dieser Dimension gesetzt, damit für die verschiedenen Fuzzy-Niveaus auch in dieser Dimension die scharfe Regel in der Größe variiert. Ließe man die unendlichen Ausdehnungen unverändert, was prinzipiell möglich ist, so würde dann die scharfe Regel mit unendlicher Ausdehnung in dieser Dimension zum Fuzzy-Niveau 1.0 nicht dem Core-Rechteck entsprechen, dass *immer* ein endliches Volumen besitzt. Die Regel heißt hier scharf, weil die Prämissenseite nun aus Intervallen mit scharfen Intervallgrenzen für die Variablen besteht.

Sei nun  $R$  die scharfe Regel, die zum Hyper-Rechteck  $H_R$  zugehörig ist.  $R$  hat eine feste Klassenzugehörigkeit  $k$ .

b) Die **Häufigkeit**  $\text{freq}(R)$  der scharfen Regel  $R$  der Klasse  $k$  ist definiert als die Anzahl von Messwerten, die in  $H_R$  liegen, geteilt durch die Anzahl der Messwerte des ganzen Datensatzes.

c) Die **Konfidenz**  $\text{conf}(R)$  der scharfen Regel  $R$  der Klasse  $k$  ist definiert als die Anzahl von Messwerten der Klasse  $k$ , die in  $H_R$  liegen, geteilt durch die Anzahl der Messwerte aller Klassen, die in  $H_R$  liegen.<sup>5</sup>

Es folgen die Ergebnisse der Konfidenz- und Häufigkeitsberechnungen auf den scharfen Regeln der verschiedenen Datensätze (siehe die folgenden Abbildungen). Die genauen numerischen Ergebnisse für die Core- und Support-Regeln findet der Leser im Anhang C. Um Kennlinien zu erhalten, wurden aus allen Trapezoiden jeweils scharfe Regeln zum selben Fuzzy-Niveau erzeugt. Die Kennlinien geben jeweils die mittleren Konfidenz- bzw. Häufigkeitswerte über alle Regeln einer Klasse wieder.

**IRIS (Fuzzy-Regeln zu den Niveaus 0.0 bis 1.0):** In der Abb. 3.12 sieht man die Zunahme der mittleren Konfidenz bei zunehmenden Fuzzy-Niveau, bei Klasse 2 bis zum Niveau 0.9. Danach haben die Regeln eine zu geringe mittlere Häufigkeit, siehe auch die Erklärung zu den Diabetes1-Daten zum Fuzzy-Niveau 0.0 bis 1.0, die noch folgt. - Die mittlere Häufigkeit der 2. und 3. Klasse nimmt kontinuierlich ab (Abb. 3.13), während für die 1. Klasse kaum eine Abnahme beobachtet werden kann; das liegt daran, dass es nur sehr wenige Regeln für diese Klasse gibt, und die Support-Bereiche fast den Core-Bereichen entsprechen.

**Cancer1 (Fuzzy-Regeln zu den Niveaus 0.0 bis 1.0):** In der Abb. 3.14 sieht man die steigende Kurve der mittleren Konfidenz für die Klasse 1. Die Kurve für die Klasse 2 steigt bis 0.1 und fällt anschließend. Die fallenden Kurven der mittleren Häufigkeit sieht man in der Abb. 3.15. Der Abfall ist bis zum Niveau 0.1 deutlich stärker. Mittlere Häufigkeit und mittlere Konfidenz verhalten sich hier komplementär. In dieser Situation kann man nur sagen, welches Niveau die sinnvollsten Regeln liefert, wenn man angibt, ob eher die Häufigkeit oder eher die Konfidenz der 1. bzw. der 2. Klasse wichtig erscheint.

**Diabetes1 (Fuzzy-Regeln zu den Niveaus 0.0 bis 1.0):** In den folgenden beiden Abbildungen (3.16 und 3.17) sind die mittlere Konfidenz und die mittlere Häufigkeit der Regeln in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten) dargestellt.

Man sieht in der Abb. 3.17 deutlich, wie mit zunehmendem Fuzzy-Niveau die mittlere Häufigkeit abnimmt. - Leider stellt sich in der Abb. 3.16 bei diesem Datensatz nicht der Effekt ein, dass die mittlere Konfidenz mit steigendem Fuzzy-Niveau zunimmt. Die Erklärung dafür ist, dass oft Regeln mit kleinerer Häufigkeit gar nicht

---

<sup>5</sup>In den folgenden Beispielen werden die Gütemaße immer auf den Testdatensätzen berechnet.

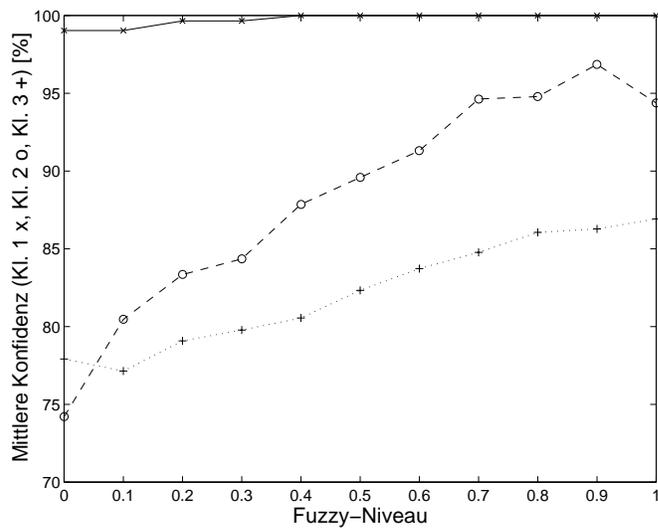


Abbildung 3.12: IRIS-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

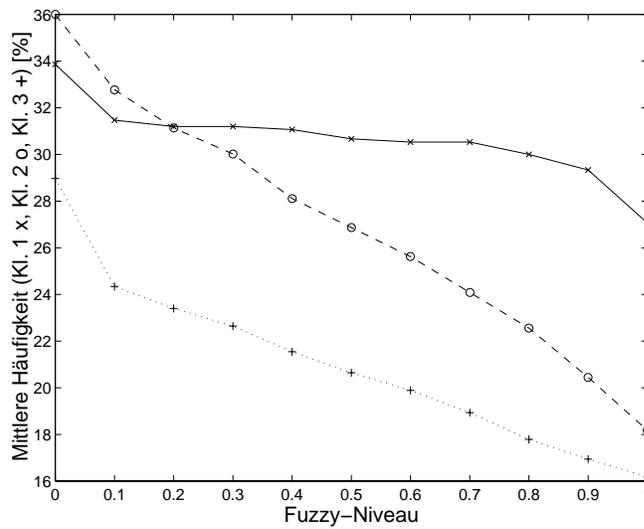
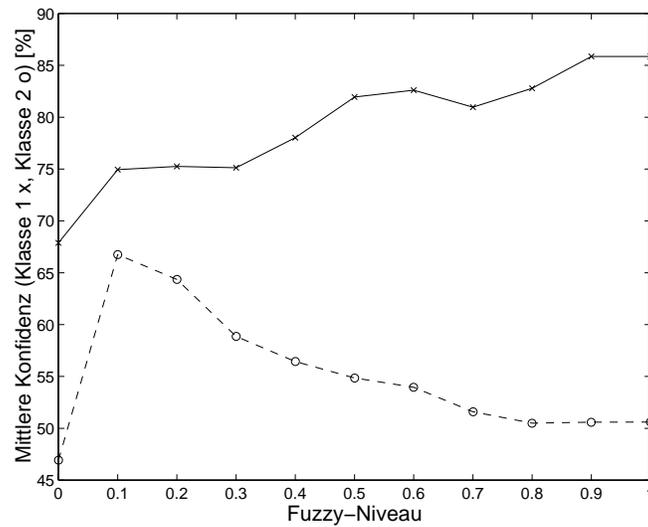
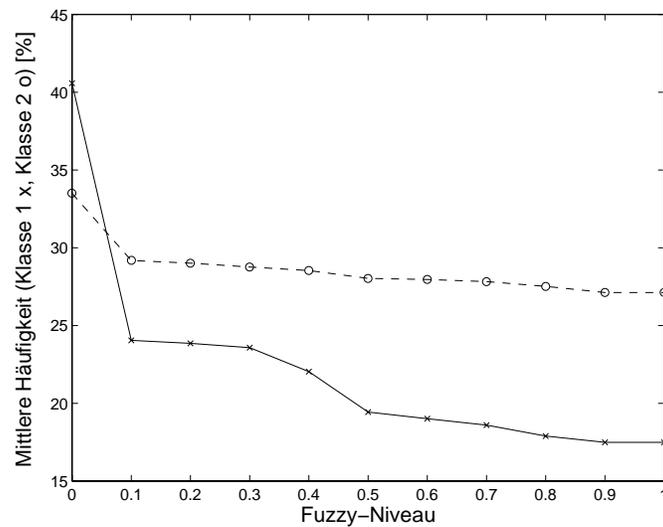


Abbildung 3.13: IRIS-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 3.14:** Cancer1-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 3.15:** Cancer1-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

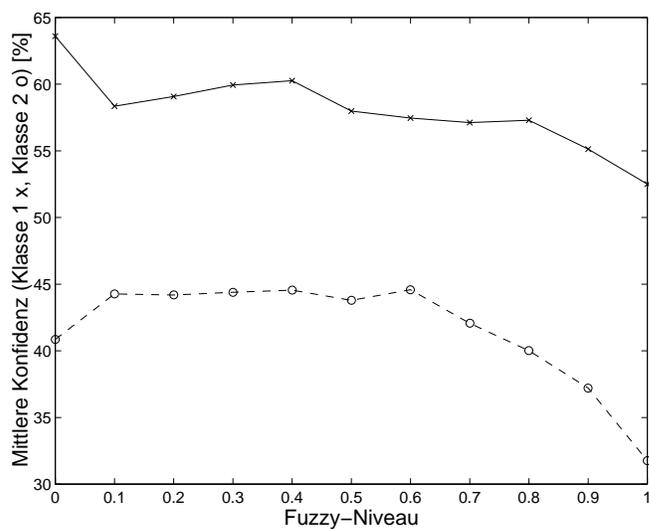


Abbildung 3.16: Diabetes1-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

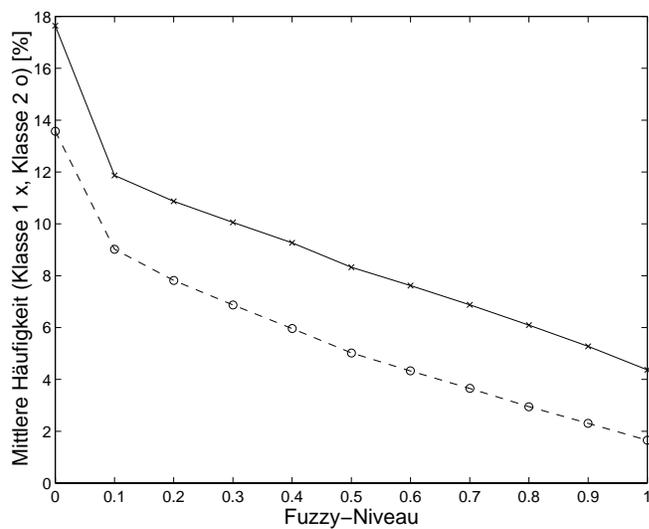
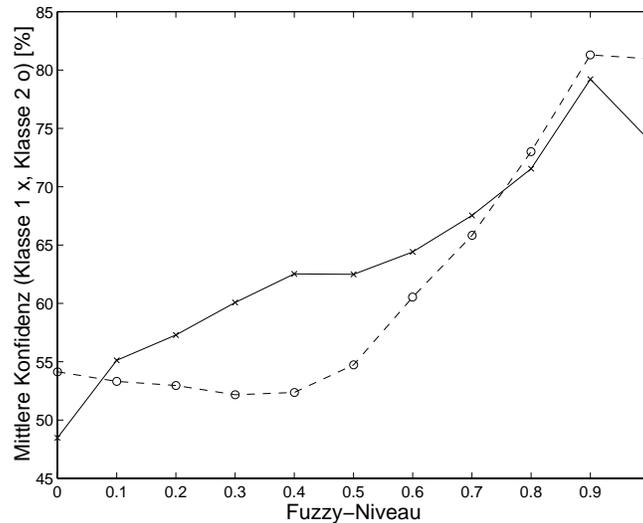


Abbildung 3.17: Diabetes1-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 3.18:** Spiral-Daten ( $nearby = 0.14$ ): Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

von den *Testdaten* getroffen werden. Setzt man dann die Konfidenz im Einzelfall auf Null, so erreicht man insgesamt keine bessere, nachweisbare Konfidenz.

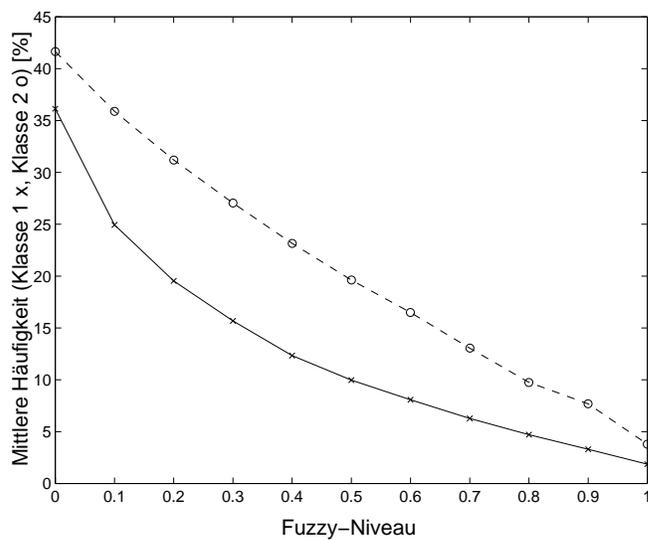
Hier heißt das, dass die Regeln für die Klasse 1 zum Niveau 0.0 (Support-Niveau) bereits diejenigen Regeln sind, die man angeben sollte, während man für die Klasse 2 je nach Wichtigkeit von Häufigkeit und Konfidenz die Regeln zwischen dem Niveau 0.1 und 0.6 wählen sollte.

**Spiraldaten, nearby-Parameter = 0.14 und = 2 (d.h. nearby-Parameter ohne Einfluss) (Fuzzy-Regeln zu den Niveaus 0.0 bis 1.0):** Bei den Spiraldaten ( $nearby = 0.14$ ) sinkt die mittlere Häufigkeit kontinuierlich (Abb. 3.19). In der Abb. 3.18 sieht man deutlich, dass die Wahl der Regeln zum Niveau 0.9 sinnvoller ist als zum Core-Niveau 1.0.

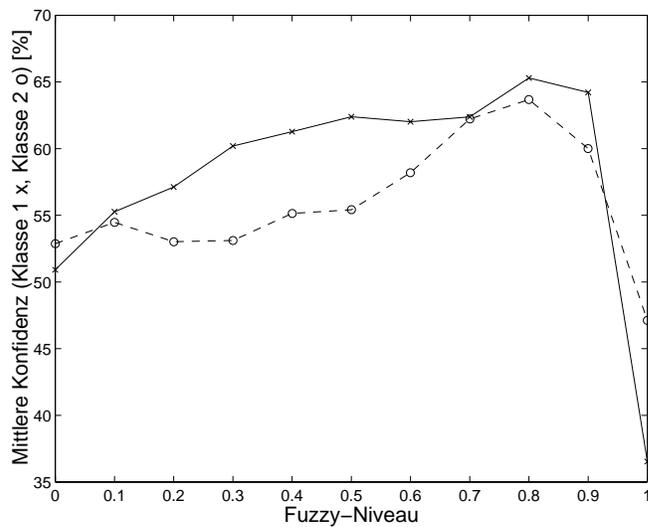
Bei den Spiraldaten (ohne *nearby*) sinkt die mittlere Häufigkeit ebenfalls kontinuierlich (Abb. 3.21). In der Abb. 3.20 sieht man deutlich, dass die Wahl der Regeln zum Niveau 0.8 sinnvoller ist als zum Core-Niveau 1.0.

Vergleicht man die Ergebnisse aus den Abb. 3.18 und 3.19 mit denen aus den Abb. 3.20 und 3.21, so stellt man fest, dass mit *nearby*-Parameter die mittlere Häufigkeit geringer ist, während man einen deutlichen Zugewinn an Konfidenz im Niveau-Bereich 0.5 bis 1.0 bekommt.

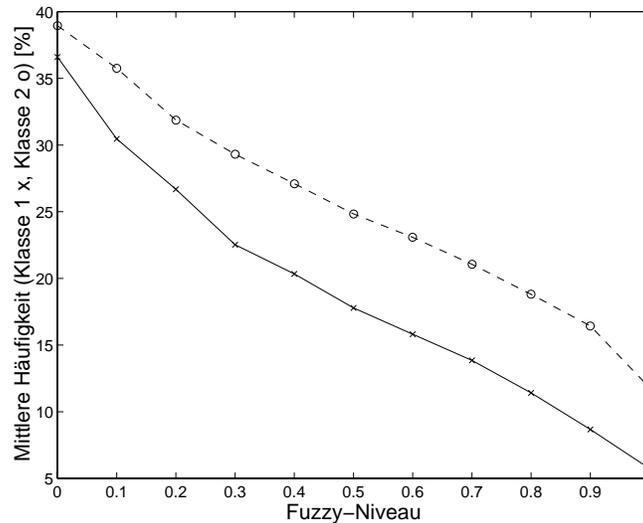
Um eine *optimale* mittlere Konfidenz zu erhalten, kann man für jede Regel separat das optimale Fuzzy-Niveau bestimmen. Beispielhaft für den Diabetes1-Datensatz erhält man für die beiden Klassen die in den Abbildungen 3.22 und 3.23 angege-



**Abbildung 3.19:** Spiral-Daten ( $nearby = 0.14$ ): Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 3.20:** Spiral-Daten ( $nearby = 2$  (ohne)): Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

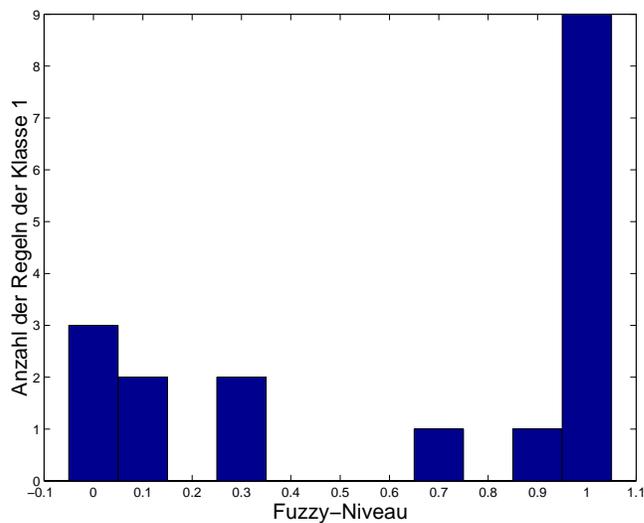


**Abbildung 3.21:** Spiral-Daten ( $nearby = 2$  (ohne)): Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

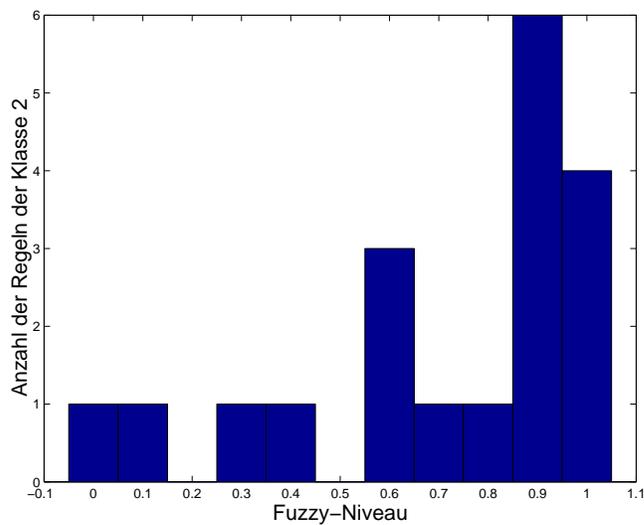
benen Verteilungen bez. des Fuzzy-Niveaus. Als mittlere Konfidenz der optimalen scharfen Regeln ergibt sich für die Klasse 1 der Wert 78.86% und für die Klasse 2 der Wert 63.77%. Das sind deutlich bessere Werte als die in der Abb. 3.16 maximal möglichen Konfidenzwerte bei einheitlicher Wahl des Fuzzy-Niveaus. Bei den anderen Datensätzen waren ebenfalls Verbesserungen zu sehen, wenngleich nicht so hohe wie bei den Diabetes1-Daten. Diese Untersuchung wird im Abschnitt 7.5 für den SystDiasThrombo3K-Datensatz durchgeführt.

**Fazit:** Insgesamt kann man feststellen, dass es aufgrund der Ergebnisse sinnvoll ist, nicht nur die Core- und die Support-Regeln zu betrachten, sondern auch die scharfen Regeln zu einem Fuzzy-Niveau zwischen 0 und 1, da man evtl. eine bessere Performanz der (scharfen) Regeln erreicht. Ein unterschiedliches Fuzzy-Niveau pro Regel verbessert die mittlere Konfidenz. Die durchgeführten Ergebnisse zeigen auch, dass der nearby-Parameter zu besseren Ergebnissen führen kann (Klassifikation, Konfidenz). Die Idealsituation, dass alle optimalen Regeln dem Fuzzy-Niveau 1.0 angehören, wird nicht erreicht.

Ergänzung: Ein Aspekt der Regelgenerierung, der mir aufgefallen ist, sei an dieser Stelle genannt. Er betrifft die *Rundung* der Variablenwerte in den Regelausgaben (hier auf zwei Stellen). Rundet man nach der Regelerzeugung, um die Ausgabe lesbarer für einen Anwender zu machen, so können sich bei zu grober Rundung die Gütemaße Konfidenz und Häufigkeit verändern, so dass sie auf den gerundeten Ergebnissen neu zu berechnen wären. Dieser technische Aspekt wird hier nicht weiter im Detail untersucht; man kann sich aber leicht Datensätze vorstellen, bei denen die Güte der Regeln durch Nachkommastellen bestimmt wird und eine Rundung die Klassifikationsgüte verschlechtert.



**Abbildung 3.22:** Diabetes1-Daten: Verteilung der optimalen scharfen Regeln der Klasse 1 in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 3.23:** Diabetes1-Daten: Verteilung der optimalen scharfen Regeln der Klasse 2 in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

### 3.5 Die Maße „Wichtigkeit“ und „Frühzeitigkeit“

Ein prinzipielles Problem der Regelgenerierung ist das Auffinden *interessanter* Regeln. Als Maße für die Beurteilung der Interessantheit haben wir bislang die Häufigkeits- und die Konfidenzmaße kennengelernt, weitere (z.B. Informationsgewinn für Entscheidungsbäume) findet man in [HH99a]. Ärzte interessieren sich immer für die *Wichtigkeit* einzelner Variablen, um auf Veränderungen der wichtigsten Variablen therapierend reagieren zu können. Einen Hinweis in diese Richtung kann die ebenso genannte, hier eingeführte Kenngröße sein, die die Dimensionalität der Regeln reduzieren hilft. In Hinblick auf die MEDAN-Zeitreihen, insbesondere für ein Frühwarnsystem, ist das erstmalige Auftreten einer Regel interessant. Hierfür wird anschließend die Kenngröße der *Frühzeitigkeit* modelliert. Bei den Experimenten wird sich herausstellen, dass die Wichtigkeit bez. des Fuzzy-RecBF-DDA-Verfahrens ein nützlicher Begriff ist. Bei der Analyse der metrischen MEDAN-Daten war allerdings der Begriff der Frühzeitigkeit nicht sehr hilfreich. Das heißt aber nicht, dass er für andere metrische Datensätze nicht doch hilfreich sein könnte.

**3.5.1 Definition: (Wichtigkeit)** Sei  $\mathcal{R}$  die Menge aller erzeugten Regeln, d.h. eine Regelmenge. Eine Variable  $W$  wird in einer Regel als irrelevant (Abk.: irr.) angesehen, wenn die Grenzen der Werte  $-\infty$  und  $\infty$  sind, also der Wert der Variablen unbeschränkt ist. Für eine fest gewählte Variable  $V$  seien  $p_1^V, \dots, p_{r_l}^V \in \mathcal{R}$  die erzeugten Regelprototypen der Klasse  $K_l$ , für die die Variable  $V$  nicht irrelevant ist.  $g_l$  sei die Wahrscheinlichkeit der Klasse  $K_l$  bez. der gesamten Datenmenge.  $conf$  bezeichne die Konfidenz einer Support-Regel. Die **Wichtigkeit bez. der Klasse  $K_l$**  (engl.: Importance) einer Variablen  $V$  in der Regelmenge  $\mathcal{R}$  wird definiert als:

$$imp_{K_l}(V) := \sum_{i=1}^{r_l} freq_{K_l}(p_i^V) \cdot conf_{K_l}(p_i^V) \cdot \frac{1}{g_l} \cdot \frac{1}{\#\{W|W \text{ nicht irr. in } p_i^V\}} \quad (3.8)$$

Prinzipiell geht man davon aus, dass eine Variable um so wichtiger ist, je öfter sie in häufigen und konfidenten Regeln verwendet wird und zwar möglichst mit wenigen anderen Variablen zusammen. Der Wert  $imp_{K_l}(V)$  kann durch die vorhandenen Überlappungen der Regelbereiche über 1 liegen, d.h. die Wichtigkeit ist nicht im Intervall  $[0, 1]$  normiert.

Interessant und bedeutend ist es anhand der Wichtigkeit Variablen auszuwählen, die für eine Regelgenerierung in Frage kommen, ohne die Klassifikationsleistung deutlich zu verringern. Kann man solche Variablen weglassen, so gewinnen die erzeugten Regeln an Klarheit aufgrund der dann niedrigeren Dimensionalität.

Nach einer Serie von Versuchen hat sich herausgestellt, dass die Auswahl derjenigen Variablen i. Allg. am sinnvollsten ist, die die höchsten Wichtigkeiten besitzen. Das ist plausibel, da diese Variablen oft in häufigeren und konfidenteren Regeln Bestandteil sind. Versuche, das Verhältnis der Wichtigkeit von Klasse 1 zur Klasse 2

Nr.	Klasse 1	Klasse 2	Summe
1	0.647	0.345	0.992
2	<b>0.822</b>	0.441	1.263 <sup>h</sup>
3	0.270	0.155	0.425
4	0.265	0.454	0.719
5	0.101	0.077	0.178
6	0.740	<b>0.568</b>	1.308 <sup>h</sup>
7	0.158	0.019	0.177 <sup>n</sup>
8	0.753	0.364	1.117
9	0.000	0.026	0.026 <sup>n</sup>

**Tabelle 3.21:** Cancer1-Daten: Wichtigkeit der Variablen, gemittelt über alle Versuchswiederholungen, bez. den Klassen „überlebt“ und „verstorben“. Der größte Wert pro Klasse ist fett gedruckt. Die beiden höchsten Werte in der Spalte „Summe“ sind mit „h“, die beiden niedrigsten mit „n“ markiert.

bei einem 2-Klassen-Problem heranzuziehen, blieben ohne Erfolg. Auch ließe sich eine solche Quotientenbildung nicht gut auf mehrere Klassen übertragen.

Interessant ist in diesem Zusammenhang folgendes Phänomen: Berechnet man die Korrelation der Werte in der Spalte „Klasse 1“ zu den Werten in der Spalte „Klasse 2“ in der Tab. 3.21, so findet man beim Cancer1-Datensatz eine hohe signifikante positive Korrelation (Wert 0.82; Signifikanzniveau 0.01). Diesen Effekt findet man auch bei den Werten in den Tabellen 5.15 (Wert 0.67; Signifikanzniveau 0.05) und 5.19 (Wert 0.70; Signifikanzniveau 0.01). D.h. eine Aufteilung der Wichtigkeit in Klassen (bez. einer Regelmenge, die vom Fuzzy-RecBF-DDA-Verfahren erzeugt wurde) ist zwar etwas genauer, aber nicht unbedingt notwendig. Man könnte auch die Summe über die beiden Klassen als Indikator der Wichtigkeit verwenden wie in [Pae]. Eine Ursache ist wohl der cover-commit-shrink-Mechanismus des Fuzzy-RecBF-DDA: Ist eine Variable wichtig für die Klassenaufteilung, so ist die Variable in Regeln beider Klassen vertreten. Eine starke Begünstigung einer Klasse findet bei einem 2-Klassen-Problem nicht statt.

Anhand des Datensatzes Cancer1 wird dem Leser die Wohldefiniertheit der Wichtigkeit erläutert [Pae]. In der Tab. 3.21 sieht man, dass die höchste Wichtigkeit für Klasse 1 von der Variable 2 angenommen wird, für die Klasse 2 von der Variablen 6. Die Variablen 5, 7 und 9 haben eher niedrige Wichtigkeiten. Die anderen Variablen haben Werte dazwischen.

Das Klassifikationsergebnis einer Regelgenerierung mit lediglich den beiden Variablen 2 und 6 ist zwar geringfügig schlechter als das mit allen 9 Variablen (die anderen Variablen waren ja auch nicht unwichtig), aber die Regeln sind als maximal zweidi-

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	93.90	1.47	91.12	95.70
Training nicht klassifiziert	0.06	0.18	0.00	0.57
Test korrekt	94.43	1.66	90.86	96.86
Test nicht klassifiziert	0.03	0.09	0.00	0.29

**Tabelle 3.22:** Cancer1-Daten: korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten (2 aus 9 Variablen, d.h. Var. 2 und 6). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	89.74	3.61	80.23	92.26
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	89.14	5.32	74.86	92.86
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 3.23:** Cancer1-Daten: korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten (2 aus 9 Variablen, d.h. Var. 7 und 9). Angaben in Prozent.

mensionale Regeln deutlich einfacher strukturiert als die bis zu 9-dimensionalen bei sonst identischer Versuchsdurchführung, vgl. Tab. 3.5 und 3.22.

Die Variablen 2 und 6 liefern das beste Klassifikationsergebnis unter allen Klassifikationen mit den  $8 + 7 + \dots + 2 + 1 = 36$  verschiedenen Kombinationen aus 2 der 9 Variablen. Eines der schlechtesten Ergebnisse ist das Ergebnis eines Versuches (wiederum 10 Wiederholungen) mit den (unwichtigeren) Variablen 7 und 9 in der Tab. 3.23 (bei sonst identischem Versuchsaufbau).<sup>6</sup> Man sieht hier deutlich die Verschlechterung der Ergebnisse bei der Wahl unwichtigerer Variablen. Ein im Mittel 4% schlechteres Klassifikationsergebnis bei prinzipiell gut zu klassifizierenden Daten zeigt hier deutliche Unterschiede.

Neben den Ergebnissen, die in den Tabellen 3.22 und 3.23 mit dem Fuzzy-RecBF-DDA erzielt wurden, wurde die Versuchsdurchführung mit *allen* möglichen Paaren zweier Variablen aus Zeitgründen mit der logistischen Regression aus SPSS 9.0 durchgeführt. Die Ergebnisse der Regression bestätigte die Tendenz: Je wichtiger die Variablen, desto besser die Klassifikation; je unwichtiger die Variablen, desto schlechter die Klassifikation.

<sup>6</sup>Da der summierte Wert der Wichtigkeit für Variable 5 fast gleich dem der Variable 7 ist, liefern Klassifikationen mit dieser Variable ähnlich schlechte Werte.

Nr.	Klasse 1	Klasse 2	Klasse 3
1	0.000	0.269	0.162
2	0.000	0.349	0.106
3	0.644	0.502	0.460
4	0.360	0.724	0.916

**Tabelle 3.24:** IRIS-Daten: Wichtigkeit der Variablen bez. den drei Klassen, gemittelt über alle Versuchswiederholungen.

Die Wichtigkeit der 4 Variablen bez. den 3 Klassen im IRIS-Datensatz ist in der Tab. 3.24 wiedergegeben. Man sieht, dass die Klasse 1 allein durch die Variablen 3 und 4 getrennt wird, Klasse 3 hauptsächlich durch die Variablen 3 und 4 und Klasse 2 zwar auch hauptsächlich durch die Variablen 3 und 4, aber auch etwas mehr durch die Variablen 1 und 2. Kürzlich wurde in [SB99] ein Informationsmaß zur Bewertung der Wichtigkeit der Eingabevariablen bez. des Klassifikationsergebnisses vorgestellt, das eine zusätzliche Gitterbildung erfordert und nicht zwischen der Wichtigkeit für verschiedene Klassen unterscheidet, was bei der hier vorgestellten Kenngröße möglich ist, falls mehr als 2 Klassen gegeben sind. Würde ich in der Tab. 3.24 die Summe der Wichtigkeiten über die Klassen berechnen, so wären eindeutig die Variablen 3 und 4 wichtiger als die Variablen 1 und 2. Dieses Ergebnis stimmt mit dem in [SB99, S. 94, Tab. 3] überein. Da die von mir definierte Wichtigkeit direkt auf den Häufigkeits- und Konfidenzwerten der Regeln arbeitet und somit ohne zusätzliche Gitterbildung berechnet werden kann und zudem Wichtigkeiten für verschiedene Klassen (bei mehr als 2 Klassen) ausgibt, kann die hier vorgestellte Kenngröße sinnvoll verwendet werden.

Die Wichtigkeit wird als sinnvolles Maß für die metrischen MEDAN-Daten in den Kapiteln 5 und 7 berechnet.

Ergänzung: Die Wichtigkeit stellt eine Möglichkeit zur Bewertung einzelner wichtiger Variablen dar, aber auch eine Möglichkeit zur Dimensionsreduzierung im Sinne einer Merkmalsauswahl (engl.: feature selection). Andere Möglichkeiten zur Merkmalsauswahl sind sog. „wrapper“, bei denen die gesamte Klassifikation und das wrapper-Verfahren wiederholt angewendet werden muss, um zu einer Auswahl zu gelangen, so z.B. bei einer Sensitivitätsanalyse. Der Aufwand dieser Verfahren ist recht hoch. Eine andere Möglichkeit sind sog. „filter“, die einer Klassifikation vorgeschaltet werden (z.B. Cluster, Informationsmaße, PCA), die aber die folgende Klassifikation oder gar die Regelstruktur nicht berücksichtigen. Einige dieser Merkmalsauswahlverfahren, insbesondere aus dem Bereich des maschinellen Lernens, werden in [DL97] verglichen. Problematisch ist für viele der genannten Verfahren der Umgang mit verrauschten bzw. überlappenden Daten. Die Verfahren haben alle auf dem einen oder anderen Datensatz Schwierigkeiten, die relevanten Merkmale zu extrahieren. Insgesamt erscheint die Möglichkeit, die Merkmalsauswahl mit dem Klassifikationsverfahren zu verbinden, so wie es mit der Definition der „Wichtigkeit“ in diesem Abschnitt

getan wurde, sinnvoll zu sein: Bereits nach einer Klassifikation kann man eine Reihenfolge der wichtigen Variablen für den Klassifikationsprozess aufstellen.

Es folgt die Definition der *Frühzeitigkeit* von Regeln und dazu ein Beispiel.

### 3.5.2 Definition: (Frühzeitigkeit)

Ausgangspunkt seien  $s$  Zeitreihen (z.B. von Variablenmessungen eines Patienten)  $Z_j$  der Länge  $t_j$ ; jede Zeitreihe  $Z_j$ ,  $j = 1, \dots, s$  gehöre zu einer Variablen  $V_j$ , die Werte aus einem Intervall annehmen kann. Die Regelmenge der Regelprototypen  $p$ , die aus den einzelnen Samples der Zeitreihe gewonnen wurde, sei  $\mathcal{R}$ . Bed. (A) bedeute in (3.9) „ein multidimensionaler Sample der Zeitreihen liegt im Support-Bereich der Regel“. Die (mittlere) **Frühzeitigkeit** einer Regel  $p \in \mathcal{R}$  (engl.: earliness) sei (mit  $\text{mean}$  als arithmetischen Mittelwertoperator) definiert als der mittlere Auftrittszeitpunkt der Samples in der multidimensionalen Zeitreihe  $Z = (Z_1, \dots, Z_s)$ , wobei die Zeitangaben vom Ende der Zeitreihe her gerechnet werden:

$$\text{earliness}(p) := \text{mean}_{\{Z \text{ ist multidim. Zeitreihe eines Patienten} \mid \text{Bed. (A)}\}} \left( \min_j \{t_j - t_{Z_j}\} \right). \quad (3.9)$$

Dabei sei  $t_{Z_j}$  derjenige Zeitpunkt, an dem der jeweilige Variablenwert in der eindimensionalen Zeitreihe  $Z_j$  zum *erstenmal* auftritt.

### 3.5.3 Beispiel (Frühzeitigkeit):

Gegeben seien zwei 2-dimensionale Zeitreihen  $Y$  und  $Z$ .  $Y$  stammt von einem gesunden (unkritischen) Patienten,  $Z$  von einem Patienten, der am 3. Tag in einen kritischen Zustand wechselt und verstirbt. Die zwei betrachteten Variablen seien der systolische Blutdruck und der pH-Wert. Das Normalwertintervall des systolischen Blutdrucks eines Patienten sei  $[100, 140]$  [mmHg], das des pH-Werts  $[7.35, 7.45]$ . Die Liegedauer des ersten Patienten sei 5 Tage, die des zweiten 6 Tage. Es seien die Zeitreihen der täglichen Messungen gegeben durch  $Y^{\text{Blutdruck}} = (120, 110, 120, 114, 110)$ ,  $Y^{\text{pH}} = (7.38, 7.39, 7.38, 7.40, 7.38)$  und  $Z^{\text{Blutdruck}} = (116, 125, 151, 160, 162, 155)$ ,  $Z^{\text{pH}} = (7.39, 7.41, 7.49, 7.50, 7.42, 7.51)$ . Eine Regel  $p$  aus einer generierten Regelmenge  $\mathcal{R}$  sei gegeben als „**if** BlutdruckSystolisch **in**  $(150, \infty)$  **and** pH **in**  $(7.44, \infty)$  **then** Klasse verstorben“. Die Regel  $p$  trifft auf den 3., 4. und 6. Tag des verstorbenen Patienten zu;  $p$  trifft auf keinen Tag des überlebenden Patienten zu. Es ist dann  $\text{earliness}(p) = (6 - 3)/1 = 3$ , da am 3. Tag vor dem letzten Tag die Regel (im Mittel über alle Patienten, bei denen die Regel mind. einmal zutrifft) zutraf. Eine weitere Regel  $q$  aus  $\mathcal{R}$  sei gegeben durch „**if** BlutdruckSystolisch **in**  $(-\infty, 142)$  **and** pH **in**  $(7.30, 7.45)$  **then** class survived“. Die Regel  $q$  trifft auf den 1. und 2. Tag des verstorbenen Patienten zu;  $q$  trifft auf alle 5 Tage des überlebenden Patienten zu, also insbesondere auch am 1. Tag. Es ist dann  $\text{earliness}(q) = ((6 - 1) + (5 - 1))/2 = (5 + 4)/2 = 4.5$ , da am 4.5-ten Tag vor dem letzten Tag die Regel (im Mittel) zutrifft.

Ideal für ein Frühwarnsystem sind häufige, konfidente Regeln, die eine hohe Frühzeitigkeit bez. der kritischen Klasse besitzen. Falls der Patient als Modellvorstellung vor dem Versterben ungewöhnliche Messwerte zeigt, kann dies durch ein solches System erkannt werden. Anwendbar ist das Maß der Frühzeitigkeit natürlich auch zur Modellierung von Systemausfällen in beliebigen Systemen, z.B. in Maschinenanlagen.

Die Berechnung der Frühzeitigkeit wird sich später nur für die symbolischen Daten der MEDAN-Daten als sinnvoll erweisen, da die Frühzeitigkeit der metrischen MEDAN-Daten keine Unterscheidung für die verschiedenen Klassen liefern wird.

### 3.6 Regelgewinnung mit dem Fuzzy-WvRecBF-DDA-Netz

Neben den im Abschnitt 3.1 genannten Ergänzungen des Fuzzy-RecBF-DDA-Verfahrens, wird hier nun eine zusätzliche, optionale Erweiterung vorgeschlagen, die

1. etwas robuster gegenüber Ausreißern ist, und zwar durch Erweiterung des Zählers  $w_i^k$ , der das Auftreten von Datenpunkten innerhalb des Support-Bereichs zählt, zu einem  $c$ -dimensionalen Vektor  $w_i^k = (w_{i,1}^k, \dots, w_{i,c}^k)$ , der für jede der  $c$  Klassen einen Zähler beinhaltet. Der Zähler  $w_{i,k}^k$  ist der *Hauptzähler*, der bereits im ursprünglichen Fuzzy-RecBF-DDA vorhanden ist. Die Zähler  $w_{i,k \neq l}^k$  sind die *Nebenzähler*, die die Ausreißer anderer Klassen zählen. Mit Hilfe der Nebenzähler kann das Einfügen eines neuen Neurons so lange verzögert werden, bis eine genügende Anzahl von anders zu klassifizierenden Daten in den Support-Bereich gefallen ist („insert criteria“). Prinzipiell kann man das Problem auch durch das Entfernen der von Ausreißern erzeugten Regeln und erneutem Training ohne die Ausreißer angehen (siehe [Ber99]), aber generell erscheint es sinnvoll, die Möglichkeit zu überprüfen, bereits während der (dann einzigen) Trainingsphase diese Ausreißerneuronen nicht entstehen zu lassen.
2. tendenziell kleinere Support-Bereiche erzeugt; da nicht mehr bei jedem anders zu klassifizierenden Datum (im Vergleich zur aktuellen Neuronklasse) ein neues Neuron eingefügt wird, bietet es sich an, bereits vorsorglich zu „schrumpfen“, allerdings nicht bis auf die entsprechende Koordinate des Datums, sondern nur um einen Teil des vollständigen Schrumpfbereichs (hier wird der Vorgang im Folgenden „weak shrink“ genannt).

Es gelten die gleichen Bezeichnungen wie für den Algorithmus 3.1.1.

**3.6.1 Algorithmus (Fuzzy-WvRecBF-DDA):<sup>7</sup>**

Zusätzlicher Parameter:  $min_{dp}$  bezeichnet das Minimum an Datenpunkten (anderer Klassen);  $s_{weak} \in ]0, 1[$  gibt an, um welchen Faktor geschrumpft wird im Falle eines weak shrink.  $w_{i,k}^k$  sei der Hauptzähler,  $w_{i,k \neq l}^k$  seien die Nebenzähler. Es wird das Training einer Epoche wiedergegeben.

1. Gewichte zurücksetzen:

**for**  $l = 1$  **to**  $c$

**for**  $i = 1$  **to**  $m_l$

$w_{i,1}^l := 0, \dots, w_{i,c}^l := 0$ ; das Neuron erhält für jede der  $c$  Klassen einen Zähler

$\lambda_{d,i,\pm} := 0$ ;

**end**

**end**

2. Muster durchgehen:

**for**  $r = 1$  **to**  $\#\mathcal{DS}$

**if** ( $p_i^k$  covers  $x$ ) **and** ( $\|z_i^k - x\|_2 < nearby$ ) **then**

3. Muster durch  $p_i^k$  überdeckt (cover), dann erhöhe den Hauptzähler:

$w_{i,k}^k := w_{i,k}^k + 1$ ;

  adjustiere die  $\lambda_{d,i,\pm}^k$ , so dass sie  $x$  überdecken;

  Gibt es einen Core-Konflikt, d.h. liegt  $x$  in einem Core-Bereich

  eines Prototypen einer anderen Klasse  $l \neq k$ , so setze alle

$\lambda_{d,i,\pm}^k := 0$ ; kann passieren, wenn ein Core-Bereich

  von einem Support-Bereich einer anderen Klasse überlappt wird.

**else**

4. Ergänzung zu cover: Anpassen der Gewichtsvektoren

**for**  $l \neq k$ ,

$U\{l\} := \{i | x \in \text{Support-Bereich von } p_i^l\}$ ;

**if**  $i \in U\{l\}$  für ein  $1 \leq i \leq m_l$  **then**

$w_{i,k}^l := w_{i,k}^l + 1$ ; erhöhe entsprechenden Nebenzähler

**end**

**end**

5. Berechne den Wahrheitswert (*true*, *false*) des insert-Kriteriums:

$insertcriteria := (\exists l \neq k : (U\{l\} \neq \emptyset \text{ and } p_i^l : \sum_{s, s \neq l} w_{i,s}^l \geq min_{dp}))$ ;

**if**  $insertcriteria = true$  **then**

  Setze für alle  $p_i^l$ , die die Schwellwertbedingung bez.  $min_{dp}$  erfüllt haben,

  die Nebenzähler zurück:

$w_{i,s \neq l}^l := 0$ ;

---

<sup>7</sup> „Wv“ steht für *Weight vector*.

6. Neues Neuron einfügen (commit):  
 $z_{m_k+1}^k := x$ ;  
 $\lambda_{d,m_k+1,\pm} := 0$ ; Core-Bereich ist nur der Zentrumspunkt  
 $\Lambda_{d,m_k+1,\pm}^k := \infty$ ; Support-Bereich ist unendlich
7. **for**  $l \neq k, 1 \leq j \leq m_l$   
 Schrumpfe  $p_{m_k+1}^k$  anhand  $z_j^l$ , d.h.  
 $\text{shrink}(p_{m_k+1}^k, z_j^l)$ ;  
**end**  
 $m_k := m_k + 1$ ;  
 $w_{m_k,k}^k := 1.0$ ; Hauptzähler  
 $w_{m_k,l \neq k}^k := 0.0$ ; Nebenzähler  
**else**
8. Noch nicht einfügen (weak shrink): Wie Schritt 9., aber:  
 berechne  $M$  wie im Algorithmus 3.1.2;  
 ( $N$  wird nicht benötigt)  
 Falls  $M$  existiert:  
 $M := s_{\text{weak}} \cdot M$ ; „vorsorgliches“ Schrumpfen endlicher Radien  
 Berechne  $\Lambda_{d,\min,\pm}$  und  $\Lambda_{d,\text{bestfinite},\pm}$  wie im Algorithmus 3.1.2;  
**if**  $\Lambda_{d,\text{bestfinite},\pm}$  existiert **then**  
 $\Lambda_{d,\pm} := \Lambda_{d,\text{bestfinite},\pm}$ ;  
**else**  
 $\Lambda_{d,\pm} := \Lambda_{d,\min,\pm}$ ;  
**end**  
**end**  
**end**
9. Konfliktneuronen nach cover (3.) und commit (6./7.) verkleinern (shrink):  
**for**  $l \neq k, 1 \leq j \leq m_l$   
**if**  $x$  liegt im Support-Bereich von  $p_j^l$  **then**  
 Schrumpfe  $p_j^l$  anhand  $x$ , d.h.  
 $\text{shrink}(p_j^l, x)$ ;  
**end**  
**end**  
**end for**  $r$

**Ergebnisse:** Es werden die Versuchsergebnisse bez. des Datensatzes „NormAusreißer“ (siehe den Anhang A) wiedergegeben, einmal unter Verwendung von Fuzzy-RecBF-DDA (I) und einmal unter Verwendung von Fuzzy-WvRecBF-DDA (II). Anschließend werden die Ergebnisse diskutiert und bewertet.

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	93.05	2.24	90.00	96.50
Training nicht klassifiziert	0.10	0.32	0.00	1.00
Test korrekt	91.95	3.17	87.00	96.00
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 3.25:** NormAusreißer-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten, Fuzzy-RecBF-DDA. Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1	4.4	0.7	3	5
Klasse 2	4.9	1.1	3	6

**Tabelle 3.26:** NormAusreißer-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum), Fuzzy-RecBF-DDA.

**I)** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 85$ , Ergebnisse siehe Tab. 3.25 (Klassifikation) und Tabellen 3.26, C.13 und C.14 im Anhang C (Regelgenerierung). Die verwendeten Testdaten kamen aus den 10 Testdurchläufen im Mittel zu 48.45% aus Klasse 1 und zu 51.55% aus Klasse 2. Die Epochenzahl betrug im Mittel 3.1 mit einer Standardabweichung von 0.32 (Minimum 3, Maximum 4).

**II)** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 85$ , Ergebnisse siehe Tab. 3.27 (Klassifikation) und Tabellen 3.28, C.15 und C.16 im Anhang C (Regelgenerierung). Die verwendeten Testdaten kamen den 10 Testdurchläufen im Mittel zu 50.00% aus Klasse 1 und zu 50.00% aus Klasse 2. Die Epochenzahl betrug im Mittel 3.1 mit einer Standardabweichung von 0.32 (Minimum 3, Maximum 4). Die Parameter sind eingestellt auf:  $\text{min}_{dp} := 2$  und  $s_{weak} := 0.5$ .

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	94.10	2.12	90.50	97.50
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	92.75	1.90	89.00	95.00
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 3.27:** NormAusreißer-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten, Fuzzy-WvRecBF-DDA. Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1	4.7	1.0	3	6
Klasse 2	4.8	0.8	4	6

**Tabelle 3.28:** NormAusreißer-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum), Fuzzy-WvRecBF-DDA.

**Bewertung der Ergebnisse:** Im Vergleich zum Fuzzy-RecBF-DDA (Tab. 3.25) erzielt die Erweiterung Fuzzy-WvRecBF-DDA (Tab. 3.27) ein ähnlich gutes Klassifikationsergebnis. Die Werte der Häufigkeit und insbes. auch der Core-Konfidenz sind höher. Bez. der Anzahl der erzeugten Regeln (Tab. 3.26 und 3.28) besteht kein auffälliger Unterschied. - Insgesamt kann man von einem etwas robusteren Lernverhalten sprechen als beim Fuzzy-RecBF-DDA, da die Ausreißer nicht sofort eine neue Regel erzeugen. Ein Versuch auf den Vorstudientdaten wird im Kap. 5 beschrieben. Allerdings sollte man die Variante nur einsetzen, wenn man einzelne Ausreißer vermutet; ist das nicht der Fall, z.B. bei sehr stark überlappenden Daten, so bringt die Variante keinen Vorteil.

### 3.7 Ähnlichkeit von Fuzzy-RecBF-DDA-Regeln

Hat man eine Regelmenge generiert, so kann man sich fragen, welche Regeln ähnlicher zueinander sind und welche unähnlicher. Präsentiert man einem Anwender Regeln, so ist es sinnvoll, möglichst zueinander unähnliche Regeln zu präsentieren. Viele zueinander ähnliche Regeln verwirren den Anwender eher. Dieser Aspekt ist insbesondere später bei den Generalisierungsregeln von Interesse (siehe die Abschnitte 4.9, 6.4 und 7.3).

Hat man mehrere Regelmengen mit unterschiedlichen Trainingsdaten generiert, so unterscheiden sich die Regelmengen. Möchte man eine dieser Regelmengen für eine Anwendung einsetzen, so stellt sich die Frage, welches die geeignetste ist. Einfach diejenige Regelmenge zu wählen, die die beste Klassifikation erlaubt, ist nicht sinnvoll, da die Güte der Klassifikation zufällig hoch sein kann aufgrund einer zu günstigen Wahl der Trainings- und Testdaten. Vorgeschlagen wird an dieser Stelle, bei der Wahl der Regelmenge auf deren Eigenschaften zu achten. Anhand der Eigenschaften wird eine *Ähnlichkeit* zwischen Regelmengen definiert. Ausgewählt wird dann für eine Anwendung diejenige Regelmenge, die im Mittel am ähnlichsten zu allen anderen ist [PAB].

Ursprünglich habe ich diese Überlegungen für die Generalisierungsregeln angestellt (Abschnitt 4.9) und anschließend auf die metrischen Regeln übertragen. Es ergeben sich Unterschiede aufgrund der unterschiedlichen Regelstruktur.

Zuerst wird die Ähnlichkeit von Regeln definiert.

### 3.7.1 Definition: (Ähnlichkeit)

a) Seien zwei Regeln  $R_1$  und  $R_2$  gegeben.  $rel_{R_i}$  sei die Anzahl der relevanten Variablen in der Regel  $R_i$ ,  $i = 1, 2$ . Relevant bedeutet, dass das zur Variable gehörende Intervall nicht gleich  $(-\infty, \infty)$  ist.  $rel_{id}$  sei die Anzahl derjenigen Variablen, die sowohl in  $R_1$  als auch in  $R_2$  relevant sind. Dann wird eine erste Ähnlichkeit definiert als:

$$sim_1(R_1, R_2) := \frac{rel_{id}}{\max\{rel_{R_1}, rel_{R_2}\}} . \quad (3.10)$$

b) Die Formel (3.10) aus a) berücksichtigt noch nicht die unterschiedlichen Intervallgrenzen. Sei  $i$  der Index, der die relevanten Variablen durchläuft, die sowohl in  $R_1$  als auch in  $R_2$  relevant sind, und seien  $(a_i, b_i)$  bzw.  $(c_i, d_i)$  die zugehörigen Intervalle bez. der  $i$ -ten relevanten Variable in  $R_1$  bzw.  $R_2$ . Dann definiere ich für  $i = 1, \dots, rel_{id}$ :

$$d_i^{(lr)}(R_1, R_2) := d_i^{(l)} + d_i^{(r)} . \quad (3.11)$$

Dabei bezeichnet  $d_i^{(l)}$  bzw.  $d_i^{(r)}$  die Abweichung bez. der linken bzw. der rechten Intervallgrenzen, definiert durch:<sup>8</sup>

$$d_i^{(l)} := \begin{cases} 0.5 & , a_i = c_i = -\infty \\ 0.5 \cdot (1 - |c_i - a_i|) & , a_i \neq -\infty, c_i \neq -\infty \\ 0 & , (a_i \neq -\infty \wedge c_i = -\infty) \vee (a_i = -\infty \wedge c_i \neq -\infty) \end{cases} \quad (3.12)$$

und

$$d_i^{(r)} := \begin{cases} 0.5 & , b_i = d_i = \infty \\ 0.5 \cdot (1 - |d_i - b_i|) & , b_i \neq \infty, d_i \neq \infty \\ 0 & , (b_i \neq \infty \wedge d_i = \infty) \vee (b_i = \infty \wedge d_i \neq \infty) \end{cases} . \quad (3.13)$$

Mit Hilfe dieser Abstände wird nun die **Ähnlichkeit** zweier Regeln wie folgt definiert:

$$sim_2(R_1, R_2) := \frac{sim_1(R_1, R_2) + \frac{\varrho}{rel_{id}} \sum_{i=1}^{rel_{id}} d_i^{(lr)}(R_1, R_2)}{1 + \varrho} . \quad (3.14)$$

Mit  $\varrho \in [0, 1]$  kann der Einfluss der Ähnlichkeit der Intervallgrenzen auf  $sim_2$  eingestellt werden.

---

<sup>8</sup>Da der  $m$ -dimensionale Datensatz für die Benutzung durch Fuzzy-RecBF-DDA auf  $[0, 1]^m$  abgebildet wurde, sind  $d_i^{(l)}$  und  $d_i^{(r)}$  wohldefiniert.  $d_i^{(l)} = 0.5$  bzw.  $d_i^{(r)} = 0.5$  stehen für *keinen* Unterschied;  $d_i^{(l)} = 0$  bzw.  $d_i^{(r)} = 0$  stehen für den maximalen Unterschied.

c) Im letzten Schritt werden  $g \in \mathbb{N}$  für den Anwender interessante Gütemaße  $G_1, \dots, G_g$  berücksichtigt, z.B. die Konfidenz und die Häufigkeit:

$$sim_3(R_1, R_2) := \frac{sim_2(R_1, R_2) + \sum_{j=1}^g \kappa_j (1 - |G_j(R_1) - G_j(R_2)|)}{1 + \sum_{j=1}^g \kappa_j} . \quad (3.15)$$

Mit Hilfe der Gewichtsparameter  $\kappa_j \in [0, 1]$ ,  $j = 1, \dots, g$  kann der Einfluss der statistischen Regelkenngrößen gesteuert werden.

d) Die **Ähnlichkeit zweier Regelmengen** wird wie bei den Generalisierungsregeln definiert, da diese Definition formatunabhängig ist. Seien  $R = \{R_1, \dots, R_m\}$  und  $\hat{R} = \{\hat{R}_1, \dots, \hat{R}_n\}$  zwei Regelmengen und  $S = (s_{jk}) := (sim(R_j, \hat{R}_k))$  die Matrix aller Ähnlichkeiten von Regeln der Regelmenge  $R$  und  $\hat{R}$  unter Verwendung eines der Ähnlichkeitsmaße  $sim_1$ ,  $sim_2$  oder  $sim_3$ . Sei noch  $c$  die Anzahl der Klassen und  $\#R_l$  die Anzahl der Regeln pro Klasse  $l$ . Dann wird die Ähnlichkeit zweier Regelmengen  $Sim(R, \hat{R})$  definiert durch die „besten“ Ähnlichkeiten von Regeln verschiedener Klassen und durch die Anzahl der Regeln in den Regelmengen:

$$sim(R, \hat{R}) := \frac{\frac{\sum_{j=1}^m \max_k \{s_{jk}\} + \sum_{k=1}^n \max_j \{s_{jk}\}}{m+n} + \sum_{l=1}^c \tau_l \cdot \left(1 - \frac{\min\{\#R_l, \#\hat{R}_l\}}{\max\{\#R_l, \#\hat{R}_l\}}\right)}{1 + \sum_{l=1}^c \tau_l} . \quad (3.16)$$

Mit Hilfe der Gewichtsparameter  $\tau_j \in [0, 1]$ ,  $j = 1, \dots, g$  kann der Einfluss der Regelanzahl pro Klasse gesteuert werden.

Berücksichtigt werden allerdings nur Werte  $s_{jk}$ , die von Regeln gleicher Klasse stammen. Die Ähnlichkeit von Regeln verschiedener Klassen wird generell auf 0 gesetzt, d.h. die Berechnung von  $Sim$  in d) vereinfacht sich, wenn die Maxima von  $s_{jk}$  in (4.14) nur innerhalb identischer Klassen gesucht werden.

e) Seien nun  $r$  Regelmengen  $R^{(1)}, \dots, R^{(r)}$  gegeben. Dann ist diejenige Regelmenge  $R^{(j)}$  **repräsentativ**, für die gilt:  $j = index(\max_{i=1, \dots, r} \{Sim_i\})$  mit

$$Sim_i = Sim(R_i) := \frac{\sum_{j=1, i \neq j}^r sim(R_i, \hat{R}_j)}{r - 1} , \quad (3.17)$$

d.h. die repräsentative Regelmenge ist diejenige, die im Mittel am ähnlichsten zu allen anderen Regelmengen ist. In [PAB] habe ich  $Sim_i$  kurz als „SPI“ (Abk. für: Similarity Placement Index, dt.: Platzierungskennzahl für die Ähnlichkeit) bezeichnet.

Alle Maße aus a) bis e) liegen zwischen 0 und 1.

Es folgt ein kleines Beispiel zum Teil c) der Definition: Sei  $R_1$  die Regel „**if V in**  $(-\infty, 1)$  **and W in**  $(0.2, 0.6)$  **then Kl. 1 with**  $freq = 0.2$  **and**  $conf = 0.7$ “ und  $R_2$  die Regel „**if V in**  $(-\infty, 0.5)$  **and W in**  $(0.2, \infty)$  **then Kl. 1 with**  $freq = 0.3$ “

Datensatz	#Versuche	repräsentatives $j$	$Sim(R_j)$	„worst“ $k$	$Sim(R_k)$
IRIS	10	9	0.5808	4	0.4192
Cancer1	10	6	0.6069	5	0.5274
Diabetes1	10	2	0.6590	3	0.6291
Spiralen*	8	4	0.7429	3	0.7164
Häufig12Vor	5	4	0.6704	5	0.6406
Wichtig17Vor	5	3	0.6077	1	0.5802

**Tabelle 3.29:** Ähnlichkeit der metrischen Regelmengen. Wiedergegeben ist die Anzahl der Versuche (#Versuche), der Index  $j$  der repräsentativen Regelmenge  $R_j$  und der Wert  $Sim(R_j)$ . Angegeben ist auch der Index  $k$  der ungeeignetsten Regelmenge, zusammen mit  $Sim(R_k) = \min_l \{Sim(R_l)\}$ . Die Werte von  $Sim(R_j)$  und  $Sim(R_k)$  unterscheiden sich nicht stark; schließlich sind alle Regelmengen auf demselben Datensatz erzeugt, wenngleich auch auf verschiedenen Trainingsmengen. \* ohne „nearby“-Parameter.

and  $conf = 0.9$ “, dann ist mit  $\rho = \kappa_1 = \kappa_2 = 0.5$   $sim_1(R_1, R_2) = 2/2 = 1$ ,  $sim_2(R_1, R_2) = (1 + (0.5/2)(0.5 + 0.25 + 0.5 + 0))/(1 + 0.5) = 1.3125/1.5 = 0.875$  und  $sim_3(R_1, R_2) = (0.875 + 0.5 \cdot 0.9 + 0.5 \cdot 0.8)/(1 + 0.5 + 0.5) = 0.8625$ .

In der Tab. 3.29 ist das Ergebnis der Berechnung der Ähnlichkeit der in den Versuchen (mit zufälliger Datenpartitionierung) erzeugten Regelmengen wiedergegeben. Die Parameter lauteten dabei:  $\rho = 0.1$ ,  $\tau_l = 0.1$  für alle Klassen,  $\kappa_1 = 0.1$  (Häufigkeit auf den Testdaten),  $\kappa_2 = 0.1$  (Konfidenz auf den Testdaten). Die Zahlenwerte sind für den Leser zwar nicht besonders interessant; sie zeigen aber, dass es dadurch möglich ist, automatisch eine repräsentative Regelmenge objektiv (im Sinne vorgegebener Kriterien) aus mehreren erzeugten auszuwählen, was manuell aufgrund der zahlreichen Informationen unmöglich wäre. Gerade im Fall von Datensätzen mit relativ wenigen Samples, wo eine Auswahl des besten Klassifikators statistisch nicht zu rechtfertigen ist, da das Ergebnis ja durch eine günstige zufällige Aufteilung der Daten in Trainings- und Testdaten zustande gekommen sein könnte, bietet das obige Vorgehen die Möglichkeit eine sinnvolle Modellwahl zu treffen. Die Ausführung auf den Klinikdatensätzen findet man im Abschnitt 7.2, speziell auf dem Datensatz „Häufig16K“ auch in [PAB].

Im Abschnitt 6.3 wird eine Clusteranalyse durchgeführt, um ähnliche, redundante Generalisierungsregeln zu erkennen. Analoge Experimente bez. der metrischen Regelgenerierung führen zu keiner Erkennung von vielen redundanten Regeln, da das Verfahren hier Regeln erzeugt, die keine besonders hohen Ähnlichkeiten aufweisen. Das ist als Vorteil des Verfahrens zu werten, da trotz der erlaubten Überlappung der Regeln sie nicht zu ähnlich werden. Bei den Generalisierungsregeln wird es später aber Sinn machen, nach ähnlichen Regeln zu clustern. Im Abschnitt 4.9 werden die entsprechenden Definitionen für die Generalisierungsregeln eingeführt.



# Kapitel 4

## Erzeugen von Generalisierungsregeln für symbolische Daten

Im Abschnitt 2.5.2 wurden einleitend die Grundlagen von Regelgenerierungsmechanismen für symbolische Daten vorgestellt, die aus dem Bereich des maschinellen Lernens stammen. Dabei wurde insbesondere zur Vorbereitung auf dieses Kapitel die Idee der Generalisierung mit mehrfacher, nicht-eindeutiger Klassenzugehörigkeit dem Paradigma der Assoziationsregelgenerierung gegenübergestellt. In diesem Kapitel wird nun ein adaptives Verfahren vorgestellt, das Generalisierungsregeln erzeugt und mit dem man anhand der berechneten Gütemaße für die erzeugten Regeln klassifizieren kann. Die Eigenschaften dieses Verfahrens werden diskutiert.

Es wird dabei *nicht* verlangt, dass identische Itemsets identischen Klassen angehören, sondern es wird von einer mehrfachen, nicht-eindeutigen Klassenzugehörigkeit ausgegangen. Es wird ebenfalls nicht verlangt, dass eine feste Anzahl von Items zu einem Itemset gehört, so dass insbesondere auch fehlende Werte toleriert werden. Auch werden im Abschnitt 4.5 (in Hinblick auf die Anwendung auf Daten des septischen Schocks) analog zum Abschnitt 3.5 die Gütemaße *Wichtigkeit* (für Items) und *Frühzeitigkeit* (für Regeln) eingeführt.

Um eine kombinatorische Explosion der erzeugten Regeln zu vermeiden, wird in den Abschnitten 4.6 und 4.7 eine heuristische Vorgehensweise vorgeschlagen. Eine grundsätzliche Motivation der Benutzung von Generalisierungsregeln für die Klassifikation im Vergleich zu den Assoziationsregeln wird im Nachhinein, nachdem alle Eigenschaften genannt wurden, im Abschnitt 4.8 gegeben. Die vergleichenden Ergebnisse im Abschnitt 6.6 belegen zusätzlich wie sinnvoll die Erzeugung von Generalisierungsregeln ist.

Im letzten Abschnitt wird schließlich die Möglichkeit einer Modellwahl auf der Basis der Ähnlichkeit von Regeln und Regelmengen geschaffen (analog zum Abschnitt 3.7). Die Inhalte dieses Kapitels findet man auch in [Pae02a].

## 4.1 Generalisierungsregeln durch Durchschnittsbildung

Die Idee des hier betrachteten Verfahrens beruht auf dem Prinzip der iterativen Durchschnittsbildung und hat somit einen mengentheoretischen Charakter. Da bei einer großen Ansammlung von Itemsets (bei gleichzeitig vielen Items innerhalb eines Itemsets) im ungünstigsten Fall Muster im Umfang der Potenzmenge der Items erzeugt werden könnten (*kombinatorische Explosion*), werden in den Abschnitten 4.6 und 4.7 heuristische Vorgehensweisen vorgeschlagen, die eine suboptimale Lösung anhand eines Auswahlkriteriums mit geringerer Laufzeit finden. In der Kombination mit dem Gütemaß *Wichtigkeit* wird es möglich, Items zu selektieren gemäß ihrer Bedeutung für die Klassifikation. Gelingt es bei einem Datensatz nicht durch die vorgestellten Methoden eine Generalisierung zu erreichen, d.h. wenige Regeln mit wenigen Attributen zu finden, so dass eine ausreichende Klassifikationsperformanz erreicht wird, so muss man sich damit abfinden, dass der Datensatz in diesem Fall sehr viele *Einzelfälle* enthält. Ob dies für die Daten der septischen Schock-Patienten zutrifft, wird in den Kapiteln 6 und 7 untersucht.

Kommen wir nun zum grundlegenden Algorithmus, für dessen Verständnis zuerst ein paar einfache Definitionen vorausgeschickt werden. Da der Algorithmus nicht der Einschränkung unterliegt, dass ein Itemset nur einer Klasse zugeordnet sein darf, werden insbesondere auch fehlerhafte Klassenzugehörigkeiten toleriert.

**4.1.1 Definition:** Seien  $I$  und  $J$  zwei Itemsets aus der Menge der Itemsets  $\mathcal{IS}$ . Von einem **Durchschnitt**  $K$ , als Symbol  $K := I \cap J$ , sprechen wir, falls jedes Item aus  $K$  in  $I$  und  $J$  enthalten ist;<sup>1</sup> er heißt **nichttrivial**, falls  $K \neq I$  und  $K \neq J$ , sowie  $K \neq \emptyset$ . Die **Länge** eines Itemsets  $I$  sei  $\#\{a|a \in I\} = |I|$ . Die **Ansammlung** erweitert den Mengenbegriff um die Häufigkeit der Itemsets, d.h. in einer Menge darf jedes Itemset nur einmal oder nicht vertreten sein; in einer Ansammlung darf es beliebig oft als Element auftreten. Eine Ansammlung von Daten ist der Ausgangspunkt einer Datenanalyse; identische Itemsets können z.B. mehrfach (mit unterschiedlichen Klassenzugehörigkeiten) auftreten.

Bevor der Regelalgorithmus und die darauf aufbauende Klassifikation beschrieben wird, soll noch eine geeignete modulare Umgebung angegeben werden, von der aus die Algorithmen aufgerufen werden. In diesem Modul werden Initialisierungen vorgenommen und die Performanz überprüft. Dabei werden die Schreibweisen aus dem Abschnitt 2.5 beibehalten.

---

<sup>1</sup>Es handelt sich um eine direkte Übertragung aus der Naiven Mengenlehre.

**4.1.2 Algorithmus: (Benchmark-Modul)**

Parameter:  $\min_{freq_{K_l}}$ ,  $\min_{freq}$ ,  $\min_{conf_{K_l}}$ ,  $\min_{wconf_{K_l}}$ ,  $\min_{diff_{freq_{K_l}}}$  (Minimum der Differenz des Maximums der  $K_l$ -Häufigkeit bez. einer Klasse und des zweitgrößten Wertes)

- lese eine Ansammlung  $\mathcal{IS}$  von Itemsets ein;
- jedes Itemset gehöre zu einer Klasse  $K_l$ , hier auch abkürzend mit  $l$  bezeichnet, setze  $\mathcal{IS}_{notl} := \mathcal{IS} - \mathcal{IS}_l$ ; ( $\mathcal{IS}_l$  bezeichnet die Ansammlung der Itemsets der Klasse  $l$ ).

- I.d.R. enthält die Ansammlung  $\mathcal{IS}$  Duplikate, d.h. identische Itemsets. Diese können eliminiert werden; zusätzlich wird dann eine Liste  $F$  geführt, die die Häufigkeiten der jeweiligen Itemsets aus  $\mathcal{IS}$  pro Klasse wiedergibt.

Aus der *Ansammlung*  $\mathcal{IS}$  wird dann die *Menge*  $\mathcal{IS}^D$  ohne Duplikate mit der zugehörigen Häufigkeitsliste  $F$  gebildet.

Bsp.: (Als Klammerindex sei die Klassenzugehörigkeit notiert.)

$$\mathcal{IS} = \{[1, 2]_1; [1, 2, 3, 4]_2; [1, 2, 3, 4]_1\} \Rightarrow \mathcal{IS}^D = \{[1, 2]; [1, 2, 3, 4]\}$$

und  $F = [1 \ 0; 1 \ 1]$  (Die Häufigkeiten werden für jede Klasse angegeben und für die verschiedenen Itemsets durch ein Semikolon abgetrennt.)

Der Einfachheit halber wird im Folgenden das  $D$  weggelassen;

es sind immer Mengen, d.h. Ansammlungen ohne Duplikate gemeint.

- Generalisierungsalgorithmus 4.1.3 (siehe dort auch die Erklärung für die Parameter):

$$(\mathcal{IS}^F, \text{level}, \text{startindexlevel}) = \text{GenDurchschnitt}(\mathcal{IS}, \text{maxlevel});$$

- Dabei enthält die Ergebnismenge  $\mathcal{IS}^F$  („F“ steht für „final“) ebenfalls keine Duplikate.

$I \in \mathcal{IS}_l^F$  heißt ein *generalisiertes Itemset* oder auch *Generalisierungsregel* der Klasse  $l$ , falls die Häufigkeit von  $I$  für die Klasse  $l$  größer 0 ist, also wenn  $F_l(I) > 0$ .

- Kalkuliere für alle  $I \in \mathcal{IS}^F$  und für alle Klassen  $l$  die folgenden Maße ( $I$  stehe abkürzend für  $I \Rightarrow K_l$  mit entsprechendem  $l$ ):

$freq_{K_l}(I)$ ,  $freq(I)$ ,  $wconf_{K_l}(I)$  und  $conf_{K_l}(I)$ , definiert im Abschnitt 2.5.2.

Kalkuliere ggf. weitere Maße, die später vorgestellt werden.

- Gib alle Itemsets (Regeln)  $I \in \mathcal{IS}_l^F$  für alle Klassen  $l$  aus, für die gilt:

$$freq_{K_l}(I) \geq \min_{freq_{K_l}},$$

$$freq(I) \geq \min_{freq},$$

$$conf_{K_l}(I) \geq \min_{conf_{K_l}},$$

$$wconf_{K_l}(I) \geq \min_{wconf_{K_l}},$$

$$|\max_l(freq_{K_l}) - \text{second}_l(freq_{K_l})| \geq \min_{diff_{freq_{K_l}}}$$

(„second“ sei der zweitgrößte Wert nach dem Maximum.)

Diese Regelmenge sei mit  $\mathcal{IS}^{F \geq} = \text{GenRule}$  bezeichnet.

Der folgende Algorithmus verwendet als Startmenge die Menge aller Itemsets, die in einem (Trainings-)Datensatz vorkommen und bildet Durchschnitte aus den Itemsets.

### 4.1.3 Algorithmus: (Generalisierungsregeln nach dem Prinzip der Durchschnittsbildung „GenDurchschnitt“)

Eingabeparameter: Menge  $\mathcal{IS}$ , maxlevel (gibt an, wie viele Level maximal generalisiert werden sollen).

Ausgabeparameter:  $\mathcal{IS}^F$  (generalisierte Regeln inkl. den Regeln aus  $\mathcal{IS}$ ), level (gibt an, wie viele Level generalisiert wurden, d.h.  $\text{level} \leq \text{maxlevel}$ ), startindexlevel (die Itemsets vom Index startindexlevel(n) bis startindexlevel(n+1)-1 sind dann aus dem Level  $n$  für  $n \geq 1$ ).

1. Initialisierung:

```
 $\mathcal{IS}_{new} := \mathcal{IS};$ 
level := 1;
startlevel(level) := 1;
endofalg := false;
```

2. erzeuge Level

**while** endofalg = false **do**

```
startlevel(level+1) :=  $\#(\mathcal{IS}_{new})+1$ ;
oldIS :=  $\# \mathcal{IS}_{new}$ ;
```

durchlaufe aktuelles Level

**for** i = startlevel(level) **to** startlevel(level+1)-2

durchlaufe Itemsets ohne die der vorhergehenden Level

**for** j = i+1 **to** startlevel(level+1)-1

```
Inter :=  $\mathcal{IS}_{new}(i) \cap \mathcal{IS}_{new}(j)$ ;
```

**if** Inter nichttrivialer Durchschnitt **and** (Inter  $\notin \mathcal{IS}_{new}$ ) **then**

```
 $\mathcal{IS}_{new}(\# \mathcal{IS}_{new} + 1) := \text{Inter}$ ;
```

**end**

**end**

**end**

3. prüfe auf Abbruch

**if** ( $\# \mathcal{IS}_{new} = \text{oldIS}$ ) **or** (level  $\geq \text{maxlevel}$ ) **then**

```
endofalg := true;
```

```
 $\mathcal{IS}^F := \mathcal{IS}_{new}$ ;
```

**else**

```
level := level + 1;
```

**end**

**end while**

## 4.2 Eigenschaften der durchschnittsbasierten Regeln

Der Abschnitt stellt die Eigenschaften der im vorhergehenden Abschnitt vorgestellten Regelgenerierung dar. Angesprochen werden im Folgenden: Terminierung, Level-eigenschaft, Optimalität, Pruning, Laufzeit, Parallelität, Negationen, Itemvektoren, fehlende Werte und metrische Daten. Ergänzt wird die Diskussion durch den Abschnitt 4.8.

### 4.2.1 Bemerkung: (Terminierung)

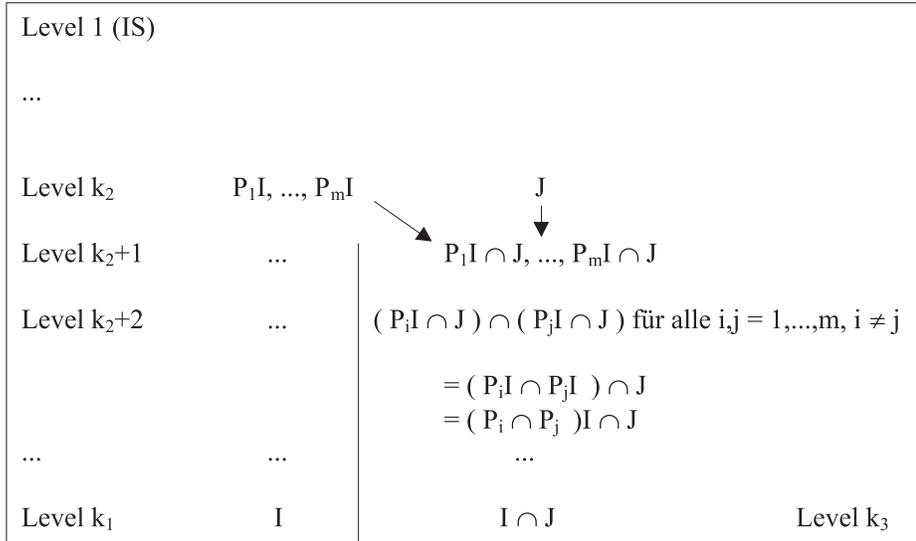
Der Algorithmus 4.1.3 geht im 1. Level von der Regelmenge  $\mathcal{IS} = \mathcal{IS}^1$  als Initialisierung aus. Er erzeugt im  $k$ -ten Level die Menge  $\mathcal{IS}^k := \{I \cap J \mid I, J \in \mathcal{IS}^{k-1} - \cup_{s=1}^{k-2} \mathcal{IS}^s\}$  durch die Bildung nichttrivialer Durchschnitte aus der Menge  $\mathcal{IS}^{k-1}$ ,  $k \geq 2$ , ohne die Itemsets vorhergehender Level aufzunehmen. Sei nun  $l_0$  die Länge des längsten Itemsets und  $l_1$  die Länge des zweitlängsten Itemsets in  $\mathcal{IS}^1$ . Die Bildung *nichttrivialer* Durchschnitte erzeugt Itemsets der Länge  $< \min\{l_0, l_1\}$ , sagen wir der maximalen Länge  $l_2$  (im 2. Level);  $l_2$  ist gegenüber  $l_0$  mindestens um 1 reduziert. Da für alle  $k \geq 2$  nur noch Itemsets der Länge  $< \min\{l_0, l_k\}$  erzeugt werden können (im  $(k+1)$ -ten Level), bedeutet das die Abnahme der Länge pro Level um mindestens 1, d.h. der Algorithmus ist spätestens im  $l_0$ -ten Level terminiert. Da das erste Level die Initialisierung darstellt, terminiert der Algorithmus nach spätestens  $l_0 - 1$  erzeugten Leveln. Dann ist  $\mathcal{IS}^F = \cup_k \mathcal{IS}^k$ .

Zusatz: I. Allg. reicht ein einziges Level nicht zur Erzeugung aller Generalisierungsregeln aus. Seien z.B. die Itemsets  $ABCD, CDEF, ABDE$  gegeben. Im 1. Level wird  $CD, ABD, DE$  erzeugt. Im 2. Level wird das zusätzliche Itemset  $D$  erzeugt. Allerdings wird man selten die maximale Anzahl von Leveln benötigen, da ja in einem Level die Länge der Itemsets um mehr als 1 abnehmen kann, z.B. erhält man aus  $ACDF$  und  $ABEGH$  das Itemset  $A$ .

### 4.2.2 Satz: (Level-eigenschaft)

Der Algorithmus 4.1.3 startet mit einer Menge von Itemsets  $\mathcal{IS}$ . Nach Beendigung des Algorithmus erhält man die erzeugte Itemsmenge (Menge der Generalisierungsregeln)  $\mathcal{IS}^F$ , wobei in dieser Menge alle erzeugten Durchschnitte liegen, inkl. der Itemsets aus  $\mathcal{IS}$ .

Die Durchschnitte werden *schichtenweise* (separat für jedes Level) gebildet, d.h. es werden nur Durchschnitte  $I_1 \cap I_2$  gebildet, für die  $I_1$  und  $I_2$  im *selben* Level liegen; Durchschnitte  $I_1 \cap I_2$ ,  $I_1 \in \mathcal{IS}^{k_1}$ ,  $I_2 \in \mathcal{IS}^{k_2}$ ,  $k_1 \neq k_2$ , werden *nicht* gebildet. Mit  $I_1 I_2$  sei die Vereinigung zweier *disjunkter* Itemsets gemeint, also  $I_1 I_2 := I_1 \uplus I_2$ .



**Abbildung 4.1:** Beweis der Leveleigenschaft: Die im Beweis verwendeten Mengen aus den verschiedenen Leveln werden als Übersicht dargestellt.  $k_3$  muss nicht gleich  $k_2$  sein.

Es wird nun bewiesen, dass durch die schichtenweise Durchschnittsbildung bereits alle Durchschnitte von Itemsets aus  $\mathcal{IS}$  gebildet werden, d.h. die folgende **Leveleigenschaft** wird bewiesen:

$$\forall (I \in \mathcal{IS}^{k_1}, J \in \mathcal{IS}^{k_2}, k_1 \neq k_2) : I \cap J \in \mathcal{IS}^F . \quad (4.1)$$

Der nun folgende Beweisgang wird in der Abb. 4.1 veranschaulicht. Gezeigt wird: Für zwei Itemsets  $I \in \mathcal{IS}^{k_1}$  und  $J \in \mathcal{IS}^{k_2}$  aus verschiedenen Leveln  $k_1 \neq k_2$  mit nichttrivialem Durchschnitt (vgl. Def. 4.1.1) wird  $I \cap J$  gebildet, d.h.  $I \cap J \in \mathcal{IS}^F$ .

Beweis:

Seien also zwei Itemsets  $I \in \mathcal{IS}^{k_1}$  und  $J \in \mathcal{IS}^{k_2}$  aus verschiedenen Leveln  $k_1 \neq k_2$  mit nichttrivialem Durchschnitt gegeben.

Dann existieren Mengen  $P_1, \dots, P_m$  ( $m \in \mathbb{N}$ ) mit  $\bigcap_{i=1}^m P_i = \emptyset$  und  $IP_1, \dots, IP_m \in \mathcal{IS}^{k_2}$ , so dass  $I$  (im Level  $k_1$ ) erzeugt wird.

Betrachte nun die nichttrivialen Durchschnitte  $P_1I \cap J, \dots, P_mI \cap J$  im  $(k_2 + 1)$ -ten Level, die nichttrivialen Durchschnitte  $(P_iI \cap J) \cap (P_jI \cap J) = (P_iI \cap P_jI) \cap J = (P_i \cap P_j)I \cap J$ ,  $i, j = 1, \dots, m, i \neq j$  im  $(k_2 + 2)$ -ten Level, usw. Da für die Mengen  $\bigcap_{i=1}^m P_i = \emptyset$  gilt, existiert ein Level  $k_3$ , in dem  $\emptyset I \cap J = I \cap J$  gebildet wird, so dass  $I \cap J \in \mathcal{IS}^F$  ist.

□

Im folgenden Satz leite ich Aussagen über die Optimalität bez. der  $K_l$ -Häufigkeit, der  $K_l$ -Konfidenz und der gewichteten  $K_l$ -Konfidenz her, die den Einsatz dieser Maße im Zusammenhang mit diesem Algorithmus rechtfertigen. Die Aussagen garantieren durch die schichtenweise Durchschnittsbildung des Alg. 4.1.3 die Erzeugung von Regeln, die den untenstehenden Optimalitätseigenschaften genügen.

#### 4.2.3 Satz: (Optimalitätseigenschaften)

Sei  $\mathcal{IS}_l^F$  die Menge der erzeugten Itemsets (Regeln), für die die Häufigkeit in der Liste  $F$  für die Klasse  $l$  größer als 0 ist (vgl. Alg. 4.1.2).<sup>2</sup> Die folgenden Maße beziehen sich alle auf eine feste, aber beliebige Klasse  $l$ . Die Aussagen sind für jede beliebige Klasse  $l$  gültig. Ist ein Itemset  $I$  in einem Itemset  $J$  echt enthalten, d.h.  $I \subsetneq J$ , so ist die Länge von  $I$  offensichtlich kürzer als die von  $J$ , d.h.  $|I| < |J|$ .

Im Folgenden sei  $J \in \mathcal{IS}_l^F$  ein „minimales“ Itemset, d.h. es gebe kein  $\tilde{J} \in \mathcal{IS}_l^F$ , für das  $\tilde{J} \subsetneq J$  gilt („Minimalitätseigenschaft“ (M)). Ist z.B.  $ABCD$  und  $ABC$  in  $\mathcal{IS}_l^F$ , so gelten die unten getroffenen Aussagen nicht für  $ABCD$ .

a) Es existiert kein  $I \notin \mathcal{IS}_l^F$ ,  $I \subsetneq J \in \mathcal{IS}_l^F$ , so dass  $\text{freq}_{K_l}(I \Rightarrow K_l) > \text{freq}_{K_l}(J \Rightarrow K_l)$  gilt, sog. „Optimalität der erzeugten Regeln bez. der  $K_l$ -Häufigkeit“. Nochmal in Worten: Es existiert kein Itemset  $I$  in der Menge  $\mathcal{IS}_l^F$  der erzeugten Itemsets (Regeln), das echt in einem erzeugten Itemset  $J$  enthalten ist und das eine höhere  $K_l$ -Häufigkeit als das Itemset  $J$  hat. Zusatz: Es gilt sogar  $\text{freq}_{K_l}(I \Rightarrow K_l) = \text{freq}_{K_l}(J \Rightarrow K_l)$ .

b) Es existiert kein  $I \notin \mathcal{IS}_l^F$ ,  $I \subsetneq J \in \mathcal{IS}_l^F$ , so dass  $\text{freq}(I \Rightarrow K_l) > \text{freq}(J \Rightarrow K_l)$  gilt, sog. „Optimalität der erzeugten Regeln bez. der Häufigkeit“. Zusatz: Es gilt sogar  $\text{freq}(I \Rightarrow K_l) = \text{freq}(J \Rightarrow K_l)$ .

c) Es existiert kein  $I \notin \mathcal{IS}_l^F$ ,  $I \subsetneq J \in \mathcal{IS}_l^F$ , so dass  $\text{conf}_{K_l}(I \Rightarrow K_l) > \text{conf}_{K_l}(J \Rightarrow K_l)$  „Optimalität der erzeugten Regeln bez. der  $K_l$ -Konfidenz“. Zusatz: Es gilt sogar  $\text{conf}_{K_l}(I \Rightarrow K_l) = \text{conf}_{K_l}(J \Rightarrow K_l)$ .

d) Es existiert kein  $I \notin \mathcal{IS}_l^F$ ,  $I \subsetneq J \in \mathcal{IS}_l^F$ , so dass  $w\text{conf}_{K_l}(I \Rightarrow K_l) > w\text{conf}_{K_l}(J \Rightarrow K_l)$  „Optimalität der erzeugten Regeln bez. der gewichteten  $K_l$ -Konfidenz“. Zusatz: Es gilt sogar  $w\text{conf}_{K_l}(I \Rightarrow K_l) = w\text{conf}_{K_l}(J \Rightarrow K_l)$ .

Beweis:

Da der Beweis des Teils a) etwas langwierig in der Ausführung ist, werden zuerst die Aussagen des Teils b), c) und d) bewiesen, die die Aussage aus a) verwenden.

---

<sup>2</sup>Man verwechsle nicht  $\mathcal{IS}$  (Startlevel) mit  $\mathcal{IS}_l^F$  (Menge der in allen Levels erzeugten Itemsets, inkl. den Itemsets aus dem Startlevel  $\mathcal{IS}$ ), deren Häufigkeit bez. der Klasse  $K_l$  größer als 0 ist.

b) Anders notiert ist  $freq(I \Rightarrow K_l) = \#\mathcal{IS}(I)$ . Dann gilt zusammen mit (2.23)

$$\begin{aligned}
\#\mathcal{IS}(I) &= \sum_{s=1}^m \#\mathcal{IS}_s(I) \\
&\stackrel{(2.23)}{=} \sum_{s=1}^m freq_{K_s}(I \Rightarrow K_s) \cdot \#\mathcal{IS}_s \\
&\stackrel{a)}{=} \sum_{s=1}^m freq_{K_s}(J \Rightarrow K_s) \cdot \#\mathcal{IS}_s \\
&\stackrel{(2.23)}{=} \sum_{s=1}^m \#\mathcal{IS}_s(J) \\
&= \#\mathcal{IS}(J) .
\end{aligned} \tag{4.2}$$

Der Zusatz ist durch die Gleichheit ebenfalls bewiesen.

c) Es gilt per definitionem (2.24):

$$conf_{K_l}(I \Rightarrow K_l) = freq_{K_l}(I \Rightarrow K_l) \cdot \frac{\#\mathcal{IS}_l}{\#\mathcal{IS}(I)} .$$

Sei nun  $I \subsetneq J$  wie in a) definiert. Dann gilt wegen a) und b)

$$\begin{aligned}
conf_{K_l}(I \Rightarrow K_l) &= freq_{K_l}(I \Rightarrow K_l) \cdot \frac{\#\mathcal{IS}_l}{\#\mathcal{IS}(I)} \\
&\stackrel{a),b)}{=} freq_{K_l}(J \Rightarrow K_l) \cdot \frac{\#\mathcal{IS}_l}{\#\mathcal{IS}(J)} \\
&= conf_{K_l}(J \Rightarrow K_l) ;
\end{aligned} \tag{4.3}$$

damit ist bereits der Zusatz bewiesen.

d) Es gilt per definitionem (nach Einsetzen der Definition für die  $K_l$ -Konfidenz) und durch Anwendung des Zusatzes von Teil a)

$$\begin{aligned}
wconf_{K_l}(I \Rightarrow K_l) &\stackrel{(2.25)}{=} \frac{freq_{K_l}(I \Rightarrow K_l)}{\sum_{s=1}^m freq_{K_s}(I \Rightarrow K_s)} \\
&\stackrel{a)}{=} \frac{freq_{K_l}(J \Rightarrow K_l)}{\sum_{s=1}^m freq_{K_s}(J \Rightarrow K_s)} \\
&\stackrel{(2.25)}{=} conf_{K_l}(J \Rightarrow K_l) .
\end{aligned} \tag{4.4}$$

Der Zusatz von Teil d) ist damit ebenfalls bewiesen.

Es folgt nun der zurückgestellte Beweis der Aussage a) des Satzes.

Level 1 (IS)	$M_1 I_{I_j}, \dots, M_n I_{I_j}$		$I_{I_{K_1}}, \dots, I_{I_{K_m}}$
...			
Level $k_3 < k_2$	[ ii) $J = I_{I_j}$ ]	Fall 2)	Fall 1) I Level $k_1$
...			
Level $k_3 = k_2$	i) $J = I_{I_j}$	I L	
...			
Level $k_3 > k_2$	[ iii) $J = I_{I_j}$ ]		
...			
beliebiges Level	$\alpha)$ I	$\beta)$ I L <sub>1</sub>	
$IS_1^F$	J	I bzw. I L <sub>1</sub> (Widerspruch)	

**Abbildung 4.2:** Beweis der Optimalität der Gütemaße: Die im Beweis verwendeten Mengen aus den verschiedenen Leveln werden als Übersicht dargestellt. In der Abb. werden die Fälle 2ii) und 2iii) in den eckigen Klammern [ ] nicht berücksichtigt.

a) Die Schwierigkeit des Beweises liegt in der Funktionsweise des Algorithmus begründet: Es können nur Durchschnitte von Itemsets gebildet werden, die im *gleichen* Level vorhanden sind. Da Itemsets Mengen darstellen, kann der Beweis im Rahmen der Mengenlehre geführt werden. Mit  $I_1 I_2$  sei wiederum die Vereinigung zweier *disjunkter* Itemsets gemeint, also  $I_1 I_2 := I_1 \uplus I_2$ . In der Abb. 4.2 sind die im Beweis verwendeten Mengen dargestellt. Der Übersicht halber werden die Fälle 2ii) und 2iii) nicht in der Abb. 4.2 berücksichtigt.<sup>3</sup>

Beweis durch Widerspruch: Angenommen, es existiert ein  $I \notin \mathcal{IS}_l^F$ ,  $I \subsetneq J$ ,  $J \in \mathcal{IS}_l^F$ , so dass  $freq_{K_l}(I \Rightarrow K_l) > freq_{K_l}(J \Rightarrow K_l)$  (\*) gilt.

Es sei  $I_J := J - I$ , also  $J = II_J$ . Wegen  $J \in \mathcal{IS}_l^F$  existieren (evtl. leere) Mengen  $M_1, \dots, M_n$ , ( $n \in \mathbb{N}$ ) mit  $M_1 II_J, \dots, M_n II_J \in \mathcal{IS}$ . Notwendig muss wegen  $J \in \mathcal{IS}_l^F$  der Durchschnitt  $\bigcap_{i=1}^n M_i = \emptyset$  sein. Deshalb existiert ein Level, in dem  $\emptyset II_J = J$  erzeugt wird.

Außerdem existieren wegen (\*) Mengen  $K_1, \dots, K_m \in \mathcal{IS}$  mit  $I \subset K_j$  ( $j = 1, \dots, m$ ,  $m \in \mathbb{N}$ ), aber  $J \not\subset K_j$  (für alle  $j$ ). Sei  $I_{K_j} := K_j - I$ , also  $K_j = II_{K_j}$  (für alle  $j$ ). Die Mengen  $II_{K_1}, \dots, II_{K_m}$  liegen also in  $\mathcal{IS}$ .

Wegen  $J \not\subset K_j$  (für alle  $j$ ) und  $J = II_J$ ,  $K_j = II_{K_j}$  gilt  $II_J \not\subset II_{K_j}$ , also insbesondere  $I_J \not\subset I_{K_j}$  (\*\*).

<sup>3</sup>Der Leser kann sich aber leicht eine entsprechende Skizze anfertigen.

Fallunterscheidung:

Fall 1)  $\bigcap_{j=1}^m I_{K_j} = \emptyset \Rightarrow$  es existiert ein Level  $k_1$ , so dass  $I$  in diesem Level als  $I\emptyset = I$  erzeugt wird, d.h. es ist  $I \in \mathcal{IS}_l^F$ . Das ist ein Widerspruch zur Annahme  $I \notin \mathcal{IS}_l^F$ .

Fall 2)  $\bigcap_{j=1}^m I_{K_j} =: L \neq \emptyset$ . Es tritt einer der drei Fälle 2i), 2ii) oder 2iii) ein, abhängig davon, in welchem Level  $k_3$  das Itemset  $J$  erzeugt wurde ( $k_3 = k_2$ ,  $k_3 < k_2$  oder  $k_3 > k_2$ );  $k_2$  sei im Folgenden dasjenige Level, in dem  $IL$  erzeugt wurde.

Fall 2i)  $k_3 = k_2$ :

$J \cap IL$  kann gebildet werden, da beide Itemsets im *gleichen* Level liegen. Unterschieden werden die beiden Fälle 2i $\alpha$ ) und 2i $\beta$ ):

Fall 2i $\alpha$ ) Es sei  $I_J \cap L = \emptyset$ . Dann ist  $J \cap IL = II_J \cap IL = I\emptyset = I$ , so dass  $I$  erzeugt wird, d.h.  $I \in \mathcal{IS}_l^F$ . Widerspruch zur Voraussetzung  $I \notin \mathcal{IS}_l^F$ .

Fall 2i $\beta$ ) Es sei  $I_J \cap L =: L_1 \neq \emptyset$ . Dann ist  $J \cap IL = II_J \cap IL = IL_1$ , so dass  $IL_1$  erzeugt wird, d.h.  $IL_1 \in \mathcal{IS}_l^F$ . Wegen  $L_1 \neq \emptyset$  ist  $I \subsetneq IL_1$ . Wegen der Voraussetzung  $I \subsetneq J$ ,  $L_1 \subset I_J \subsetneq J$  und  $I \cap I_J = \emptyset$  ist aber  $IL_1 \subsetneq J$ . Die Beziehung  $IL_1 \subsetneq J$  ist aber ein Widerspruch zur Minimalitätsforderung (M).

Beweis des Zusatzes: Da wegen  $|I| < |J|$  die  $K_l$ -Häufigkeit von  $I$  offensichtlich nicht kleiner sein kann als die  $K_l$ -Häufigkeit von  $J$ , gilt zusammen mit der Optimalitätsaussage aus Teil a) sogar immer  $\text{freq}_{K_l}(I \Rightarrow K_l) = \text{freq}_{K_l}(J \Rightarrow K_l)$ .

Die Beweise für die Fälle 2ii) ( $k_3 < k_2$ ) und 2iii) ( $k_3 > k_2$ ) lassen sich grundsätzlich analog führen. Aufgrund der nun *unterschiedlichen* Level sind die Beweise technisch aufwendiger zu führen. Sie werden deshalb als Ergänzung formuliert.

Ergänzung:

Fall 2ii)  $k_3 < k_2$ :

$J \cap IL$  kann *nicht* direkt gebildet werden, da beide Itemsets in *unterschiedlichen* Leveln liegen.

Im Level  $k_3$  ist  $J = II_J$  vorhanden. Außerdem liegen im Level  $k_3$  die Mengen  $ILN_1, \dots, ILN_p$  ( $p \in \mathbb{N}$ ) mit  $\bigcap_{i=1}^p N_i = \emptyset$ , da im Level  $k_2$   $IL$  entstehen muss.

Unterschieden werden analog zum Fall 2i) die beiden Fälle 2ii $\alpha$ ) und 2ii $\beta$ ):

Fall 2ii $\alpha$ ) Es sei  $I_J \cap LN_i = \emptyset$  für ein  $i \in \mathbb{N}_p \Rightarrow J \cap ILN_i = II_J \cap ILN_i = I\emptyset = I$ , d.h.  $I$  wird erzeugt, also  $I \in \mathcal{IS}_l^F$ . Widerspruch zur Voraussetzung  $I \notin \mathcal{IS}_l^F$ .

Fall 2ii $\beta$ ):  $\nexists i \in \mathbb{N}_p$ , so dass  $I_J \cap L N_i = \emptyset$  gilt, also  $I_J \cap I L N_i =: L_i \neq \emptyset$  (für alle  $i \in \mathbb{N}_p$ )  $\Rightarrow I L_i$  werden für alle  $i$  erzeugt. Dann gilt für alle  $i$ : Es ist  $I \subsetneq I L_i$ . Wegen  $I \subsetneq J$ ,  $L_i \subset I_J \subsetneq J$  und  $I \cap I_J = \emptyset$  gilt aber auch  $I L_i \subsetneq J$ . Widerspruch zur Minimalitätsforderung (M).

Fall 2iii)  $k_3 > k_2$ :

$J \cap I L$  kann wiederum *nicht* direkt gebildet werden, da beide Itemsets in *unterschiedlichen* Leveln liegen.

Im Level  $k_2$  ist  $I L$  vorhanden. Außerdem liegen im Level  $k_2$  die Mengen  $I I_J Q_1, \dots, I I_J Q_q$  ( $q \in \mathbb{N}$ ) mit  $\bigcap_{i=1}^q Q_i = \emptyset$ , da im Level  $k_3$   $I I_J = J$  entstehen muss.

Unterschieden werden wiederum analog zum Fall 2i) die beiden Fälle 2iii $\alpha$ ) und 2iii $\beta$ ):

Fall 2iii $\alpha$ ): Es sei  $I_J Q_i \cap L = \emptyset$  für ein  $i \in \mathbb{N}_q \Rightarrow J \cap I L = I I_J Q_i \cap I L = I \emptyset = I$ , d.h.  $I$  wird erzeugt, also  $I \in \mathcal{I S}_l^F$ . Widerspruch zur Voraussetzung  $I \notin \mathcal{I S}_l^F$ .

Fall 2iii $\beta$ ):  $\nexists i \in \mathbb{N}_q$ , so dass  $I_J Q_i \cap L = \emptyset$  gilt, also  $I_J Q_i \cap L =: L_i \neq \emptyset$  für alle  $i \in \mathbb{N}_q$ . Es wird nun ein Beweisschritt analog zum Satz 4.2.2 verwendet: Im Level  $k_3$  existieren die Mengen  $Q_i I I_J$  und  $I L$  für alle  $i$ . Dann existieren auch die Durchschnitte  $Q_i I I_J \cap I L = (Q_i I_J \cap L) I$ . Wegen  $\bigcap_{i=1}^q Q_i = \emptyset$  gibt es ein Level  $k_4 > k_3$ , in dem  $(\emptyset I_J \cap L) I = (I_J \cap L) I$  entsteht.

Es ist  $(I_J \cap L) I \subsetneq I_J I = J$ , was wie folgt begründet werden kann: Es genügt  $I_J \cap L \subsetneq I_J$  zu zeigen, was aber bereits gilt, wenn  $I_J \subsetneq L$  ist. Im Fall 2 und damit für alle Unterfälle gilt  $L = \bigcap_{j=1}^m I_{K_j}$ . Für alle  $j$  gilt aber (\*\*\*)  $I_J \subsetneq I_{K_j}$ , so dass  $I_J \subsetneq L$  folgt.

Da also  $(I_J \cap L) I \subsetneq J$  entsteht, ergibt sich ein Widerspruch zur Minimalitätsforderung (M). □

**4.2.4 Folgerung:** Nach Satz 4.2.3 ist es unnötig, Regeln zu betrachten, für die gilt:  $I \notin \mathcal{I S}_l^F$ ,  $I \subsetneq J \in \mathcal{I S}_l^F$ . – Es macht also z.B. keinen Sinn für  $\mathcal{I S}_l^F = \{ABC, BCD, BC\}$  zusätzlich die Regel  $B$  zu betrachten.  $B \notin \mathcal{I S}^F$  ist weder häufiger,  $K_l$ -häufiger, konfidenter, noch gewichtet konfidenter als  $BC \in \mathcal{I S}^F$ .  $B$  ist zwar auch kürzer als  $BC$ , aber auch unspezifischer:  $B$  trat nicht ohne  $C$  in den Itemsets auf. Dieser Sachverhalt wird noch einmal detailliert im Abschnitt 4.8 aufgenommen.

Ein Nachteil des Algorithmus ist die kombinatorische Explosion (vgl. Bem. 4.2.6), falls sehr viele Durchschnitte (von sehr vielen Itemsets) pro Level neu erzeugt werden, die in ihrer Länge dann aber eher deutlich kleiner sind als die Itemsets, aus denen die Durchschnitte gebildet wurden. Vorteilhaft sind die  $K_l$ -Häufigkeit- und die  $K_l$ -Konfidenz-Optimalität und natürlich die Terminierung des Algorithmus. Hat man ein gut generalisierbares Problem, kann man davon ausgehen, dass nicht sehr viele Itemsets erzeugt werden, diese dann aber relativ lang sind.

#### 4.2.5 Bemerkung: (Pruning)

Interessant ist die Frage, ob man *während* der Generalisierung, Regeln mit zu geringer ( $K_l$ -)Häufigkeit oder zu geringer  $K_l$ -Konfidenz aus dem Vorgang entfernen kann, um die kombinatorische Explosion zu mildern, allerdings *ohne* die Qualität der erzeugten Regeln zu mindern. Die Maße Häufigkeit oder Konfidenz müssten dann online mitgeführt und mitberechnet werden, d.h. die Maße müssten bez. dem Durchschnittsoperator monoton sein, um ein Pruning zu ermöglichen. Offensichtlich ist das für die ( $K_l$ -)Häufigkeit nicht ohne Weiteres möglich, da man zu keinem Zeitpunkt weiß, welche Häufigkeiten und welche Itemsets noch auftreten werden, die die Maße verändern können. Die Itemsets haben schließlich unscharfe Klassenzugehörigkeiten. Auch bez. der  $K_l$ -Konfidenz ist das nicht möglich, da man auch hier nicht weiß, welche Itemsets mit welcher Klassenzugehörigkeit noch auftreten und die  $K_l$ -Konfidenz positiv oder negativ verändern. Ein Assoziationsregelmechanismus hat hier scheinbar Vorteile, da man nach der Häufigkeit prunen kann. Allerdings müssen dafür auch sehr viele Kombinationen erzeugt werden, die hinterher wieder geprunt werden. Es folgt, dass man die kombinatorische Explosion nur *heuristisch* mildern kann, indem man nicht alle möglichen Muster betrachtet. Da konfidente Regeln nicht häufig und häufige Regeln nicht konfident sein müssen, bleibt als erste einfache Möglichkeit zunächst nur eine zufällige Auswahl der Regeln, z.B. kann man im 1. Level nur  $(100-x_1)\%$ ,  $x_1 \in ]0, 100[$  der Itemsets betrachten, aus dem 2. Level nur  $(100-x_2)\%$ ,  $x_2 \in ]0, 100[$  der in dieser Schicht erzeugten Itemsets, usw. Das ist sinnvoll, da man zur Klassifikation nicht unbedingt alle (optimalen) Regeln benötigt. Der so veränderte Algorithmus 4.1.3 wird im Anschluss wiedergegeben:

0. Permutation der Reihenfolge der Itemsets  
lege *Auswahlkriterium* AK fest;

1. Initialisierung:

$\mathcal{IS}_{new} := \mathcal{IS}$ ;

level := 1;

startlevel(level) := 1;

endofalg := false;

2. erzeuge level

**while** endofalg = false **do**

Kalkuliere gemäß AK  $\mathcal{IS}_{new}$  als

$\mathcal{IS}_{new} = \{I \in \mathcal{IS}_{new} | AK(I) = true\}$ ;

startlevel(level+1) := length( $\mathcal{IS}_{new}$ )+1;

oldIS :=  $\mathcal{IS}_{new}$ ;

... (weiter wie im Algorithmus 4.1.3);

Besonders sinnvoll ist eine zufällige Auswahl von weniger als 50% der Itemsets im 1. Level; so kann man immer 50% der Daten als *Testmenge*  $\mathcal{IS}_{test}$ , unabhängig von den Trainingsdaten, verwenden. Das ist notwendig, um Zufallseffekte aufgrund zu niedriger Fallzahl zu vermeiden, die zu zufällig guten Ergebnissen führen.

**Auswahlkriterien:** Auswahlkriterien werden im Abschnitt 4.6 noch einmal aufgegriffen. Insbesondere wird dort das in der nächsten Bemerkung diskutierte Laufzeitverhalten, das bei vollständiger Itemsetgenerierung zu hoch sein kann, verbessert.

#### 4.2.6 Bemerkung: (Laufzeitverhalten und Parallelisierung)

Die Laufzeit wird im Wesentlichen von vier Einflussfaktoren bestimmt:

1. Alle Level müssen bearbeitet werden.
2. Pro Level müssen die Itemsets betrachtet werden.
3. Die noch nicht bearbeiteten Itemsets im Level müssen zur Durchschnittsbildung durchlaufen werden (Betrachtung der einzelnen Items innerhalb eines Itemsets).
4. Für die Durchschnitte muss in der Menge der vorhandenen Itemsets überprüft werden, ob sie schon vorhanden sind. Dieser Schritt kann bei einer sehr großen Anzahl von Datensätzen problematisch sein, wenn die Itemsets unsortiert vorliegen. Als Verbesserung könnte man die Datensätze eines Levels zuerst nach der Länge sortieren und eine Zeigerliste mit einem Startzeiger pro Länge anlegen und dann die Itemsets gleicher Länge in einer lexikographischen Ordnung halten (z.B.  $A < C < F \mid AB < BC < BD \mid BCE < BDF < CDE$ ).

Sei nun  $n$  die Anzahl der Itemsets und  $p$  die maximale Anzahl der Items in einem Itemset. Normalerweise ist  $n \gg p$ ; allerdings sind bei den ausgewählten symbolischen Klinikdaten nur 362 Patienten und maximal 96 Attribute vorhanden, so dass dort ausnahmsweise nur  $n > p$  gilt. I. Allg. gibt es maximal  $p - 1 \ll n$  Level-Durchläufe (Punkt 1), in praktischen Beispielen eher deutlich weniger.

Sei  $M_p$  eine Menge mit  $p$  Elementen (Items). Ausgehend von  $n$  Itemsets im 1. Level können maximal  $a_1 := \min(1/2 \cdot n(n-1), \#\mathcal{P}(M_p) - n)$  neue Itemsets erzeugt werden. Im 2. Level können dann maximal  $a_2 := \min(1/2 \cdot a_1(a_1 - 1), \#\mathcal{P}(M_{p-1}) - (n + a_1))$  Itemsets erzeugt werden. Wegen  $n \gg p$  wird am Anfang die Erzeugung in jedem Level quadratisch beschränkt sein, am Ende durch das kleinere  $s$  in  $M_s$  eher durch die noch erzeugbaren Itemsets aus wenigen Items. I. Allg. wird vorausgesetzt, dass initial  $n \ll \#\mathcal{P}(M_p)$  ist, da ansonsten nicht mehr viele Itemsets erzeugt werden könnten, der Generalisierungsprozess also überflüssig wäre. In diesem Zusammenhang wird im Abschnitt 4.6 ein sog. *Generalisierungsindex* eingeführt mit dem man die Generalisierungsfähigkeit vorgegebener Daten quantitativ bestimmen kann.

Es bleibt für ein erzeugtes Itemset die Überprüfung, ob es bereits erzeugt wurde (Punkt 4). In einem sehr ungünstigen und sehr unwahrscheinlichen Fall müssten im  $l$ -ten Level maximal  $\mathcal{O}(n^{2l})$  Itemsets überprüft werden, im günstigsten Fall kein Itemset. Insgesamt erhält man eine polynomielle Laufzeitordnung, deren Höhe im Mittel schwer abzuschätzen ist, die aber praktisch bei größeren Datensätzen und vielen Items auf jeden Fall zu hoch ist. Das ist eine deutliche Schwäche des Algorithmus.

Auf eine Analyse im Mittel wird hier nicht eingegangen, da später im Abschnitt 4.6 heuristisch die Laufzeit des Algorithmus im Mittel entscheidend verringert werden kann.

Neben der heuristischen Reduzierung der Laufzeit kann auch eine Parallelisierung des Algorithmus sinnvoll sein; es können z.B. Durchschnitte auf verschiedenen Prozessoren generiert werden, wobei dann der Schritt 2 im Algorithmus 4.1.3 synchronisiert werden muss.

#### 4.2.7 Bemerkung: (Unschärfe Itemsets)

Falls für ein Itemset die Klassenzugehörigkeit a-priori nicht eindeutig feststeht, d.h. ein Eintrag in der Häufigkeitsliste  $F$  ein Einheitsvektor ist mit einem 1-Eintrag bei der entsprechenden Klassenzugehörigkeit, so kann ohne jede Änderung des Algorithmus auch mit einer nicht-eindeutigen, unscharfen Klassenzugehörigkeit gestartet werden, da er dafür konzipiert wurde.

#### 4.2.8 Bemerkung: (Verwendung von Negationen)

Interessant kann auch die Verwendung *negativer* Items sein, z.B. nichtA (andere Schreibweise  $\bar{A}$ ); man nehme z.B. einen Außerirdischen vom Planeten „General“, der versucht mit Hilfe von Generalisierungsregeln das Fahr- und Stopp-Verhalten der Autofahrer oder Radfahrer zu ergründen.<sup>4</sup> Dann wird er die Itemsets  $\{\text{ROT}\}$ ,  $\{\text{ROT, GELB}\}$ ,  $\{\text{GELB}\}$ ,  $\{\text{GRÜN}\}$  und  $\{\text{GRÜN, HUPE}\}$ <sup>5</sup> mit den Klassen STOPP und FAHREN zur Generalisierung aussuchen, da er diese beobachtet hat. Er wird (wahrscheinlich) herausbekommen, dass Autofahrer bei GRÜN fast immer fahren (bis auf Tagträumer) und das Itemset  $\{\text{GRÜN, HUPE}\}$  immer bedeutet, dass Autofahrer fahren.  $\{\text{ROT, GELB}\}$  und  $\{\text{GELB}\}$  allein ist nicht so eindeutig.  $\{\text{ROT}\}$  bedeutet fast immer STOPP, aber da ROT auch in  $\{\text{ROT, GELB}\}$  enthalten ist, ist  $\{\text{ROT}\}$  allein auch nicht ein wirklich konfidentes Itemset für STOPP.

Die Generierung konfidenterer oder einfacherer Regeln gelingt durch die Hinzunahme der Negationen nichtROT, nichtGELB und nichtGRÜN. So wäre ein Itemset  $\{\text{ROT, nichtGELB, nichtGRÜN}\}$  ein konfidenteres Itemset für STOPP bei Autofahrern, da  $\{\text{ROT, GELB}\}$  ausgeschlossen wird.  $\{\text{nichtGRÜN}\}$  hat die gleiche Bedeutung wie die beiden Itemsets  $\{\text{ROT}\}$  und  $\{\text{GELB}\}$ , d.h. das eine Item nichtGRÜN ersetzt die beiden Items ROT, GELB. Bei Radfahrern wird der Außerirdische wahrscheinlich keine eindeutigen Hinweise auf STOPP und FAHREN bekommen.

Ein Nachteil der Verwendung der Negationen ist die Erhöhung der kombinatorischen Explosion der Regeln durch die Verlängerung der Itemsets.

<sup>4</sup>Es gibt hier zwar keine echten generalisierten Itemsets, aber die entsprechenden Gütemaße wie Häufigkeit oder Konfidenz kann er ausrechnen.

<sup>5</sup>Die Hupe gehört zwar nicht zur Ampel, aber ein Außerirdischer mag vielleicht denken, dass die Ampel hupt.

**4.2.9 Bemerkung: (Itemvektoren)**

Geht man von einer festen Reihenfolge der Items in den Itemsets aus (mit einer festen Länge), so sprechen wir sinnvoller von *Itemvektoren* und ggf. von einer Menge  $\mathcal{IV}$  von Itemvektoren. Das können z.B. Zeitsequenzen sein, die innerhalb eines Zeitfensters festgehalten wurden oder Transaktionen einer Datenbank o.ä. Als Symbol verwenden wir  $(1, 2, 3)$  statt  $[1, 2, 3]$  oder  $\{1, 2, 3\}$ , also:  $\{1, 2, 3\} = \{2, 3, 1\}$ ,  $[1, 2, 3] = [2, 3, 1]$ , aber  $(1, 2, 3) \neq (2, 3, 1)$ .

Jeder der  $n_d, d = 1, \dots, D$ , auftretenden Zahlen (stellvertretend für ein Item) kann man bez. jeder Dimension  $d$  des Itemvektors eindeutige Zahlen  $s_1^d, \dots, s_{n_d}^d, s_i^d \neq s_j^d$  zuweisen im Sinne einer Bijektion  $\mathcal{IV} \rightarrow \mathcal{IS}$  gemäß  $I = (x_1, \dots, x_D) \in \mathcal{IV} \Leftrightarrow J = \{s_{i_1}^1, \dots, s_{i_D}^D\} \in \mathcal{IS}$ .  $x_d$  ist  $i_d$ -tes Item in der Dimension  $d$  für eine festgelegte Reihenfolge. Somit kann der Generalisierungsalgorithmus für Itemsets auch für Itemvektoren angewendet werden.

Bsp.: Sei  $\mathcal{IV} = \{(1, 2, 1), (1, 2, 2), (2, 2, 1)\}$ , dann ist (mit der natürlichen Reihenfolge auf  $\mathbb{N}$ )  $\mathcal{IS} = \{[1, 3, 4], [1, 3, 5], [2, 3, 4]\}$ , da der 1 in der ersten Dimension eine 1, der 2 in der ersten Dimension eine 2, der 2 in der zweiten Dimension eine 3, der 1 in der dritten Dimension eine 4 und der 2 in der dritten Dimension eine 5 zugewiesen wurde.

Durch diese Bijektion ist sichergestellt, dass beim Generalisieren nur in der jeweiligen Dimension generalisiert werden kann.

**4.2.10 Bemerkung (Fehlende Werte):**

Der Umgang mit fehlenden Werten stellt für den vorgestellten Mechanismus, der auf einer mengenbasierten Darstellung beruht, kein Problem dar; es werden einfach nur die vorhandenen Items ausgewertet. Bei den symbolischen MEDAN-Daten ist mit fehlenden Werten zu rechnen, da Operationen o.ä. in den Patientenakten zwar dokumentationspflichtig sind, aber vermeintlich Unwichtigeres aus Zeitgründen u.U. doch nicht so genau dokumentiert wird. So ist eine Verfälschung der Ergebnisse durch fehlende Werte möglich; diese wird aber eher gering ausfallen.

**4.2.11 Bemerkung: (Metrische Daten)**

Prinzipiell ist es auch möglich, metrische Daten mit einem Generalisierungsalgorithmus zu generalisieren. Als Vorverarbeitung ist dann eine Diskretisierung der Daten nötig, z.B. durch Gitterbildung. Allerdings kann eine (unüberwachte) Diskretisierung evtl. die tatsächliche Klassengrenze schlecht widerspiegeln, da die Klassifizierung nicht von der Dichte der Daten abhängig ist, vgl. die Abb. 2.17. Ein weiterer Nachteil ist die große Anzahl von Ausgangsregeln vor der Generalisierung, die durch die Diskretisierung entsteht. Dadurch entsteht eine sehr große Anzahl von Generalisierungsregeln. Nach der Generalisierung stellt sich die Frage, wie man die zahlreichen erzeugten (evtl. nur in wenigen Dimensionen generalisierten) Generalisierungsregeln zu „größeren“ zusammenfasst. Experimente auf den IRIS-Daten (auch Cancer1- und Diabetes1-Daten) zeigen so deutliche Vorteile bez. der Anzahl und der Güte der

Regeln zugunsten des Fuzzy-RecBF-DDA-Regelgenerators, so dass metrische Werte alleine nicht mit den hier genannten Generalisierungsverfahren verarbeitet werden sollten. Die Verarbeitung metrischer Daten durch Assoziationsregeln wird in [SA96] beschrieben; es gelten aber die gleichen grundsätzlichen Überlegungen.

Die Kombination von symbolischen und metrischen Regelgeneratoren wird im Abschnitt 2.5.3 diskutiert und im Abschnitt 7.4 auf die Klinikdaten angewendet.

### 4.3 Klassifikation mit einer symbolischen Regelmengenge

Der folgende Algorithmus erlaubt eine Klassifikation der symbolischen Daten unter Verwendung einer erzeugten Regelmengenge. Er verwendet einen Winner-takes-all-Mechanismus und die im Abschnitt 2.5.2 eingeführte gewichtete Konfidenz.

#### 4.3.1 Algorithmus: (Klassifikation mit Generalisierungsregeln)

Parameter:  $\mathcal{IS}$  (alle Itemsets, die klassifiziert werden sollen),  $\mathcal{IS}^{F \geq}$  (generierte Regelmengenge, die evtl. bez. der Gütemaße gefiltert wurde),  $\text{mindiff}$  (Parameter, der angibt, inwiefern sich die mittlere Trainings-Häufigkeit bez. der ausgewählten Klasse mind. von der anderer Klassen unterscheiden muss). Man beachte, dass der Klassifikator *nur* mit den Trainingsdaten erzeugt wurde. Man erhält dadurch realistische Aussagen bez. der Verwendbarkeit der Ergebnisse, insbesondere durch Wiederholung der Verfahren mit unterschiedlichen Trainingsdaten.

1. erstelle für alle Itemsets auf der Testmenge  
die Mengen aller „zutreffenden“ gewichteten  $K_l$ -Konfidenz-Werte  
**for all** Itemsets  $I \in \mathcal{IS}_{test}$  **do**  
  für alle Klassen  $l$  sei  $WCONF_l := \emptyset$ ;  
  **for all** Itemsets (rules)  $J \in \mathcal{IS}^{F \geq}$  **do**  
    **if**  $J \subset I$  **then**  
       $WCONF_l = WCONF_l \cup \{wconf_{K_l}(J)\}$ ;  
    **end**  
  **end**  
**end**
2. berechne die maximale gewichtete  $K_l$ -Konfidenz pro Klasse;  
diese sei  $\text{max}WCONF_l$ .
3. Es sei  $\text{maxindex} = \text{index}(\text{max}_l\{WCONF_l\})$  und  
 $\text{secondindex} = \text{index}(\text{max}_{l, l \neq \text{maxindex}}\{WCONF_l\})$ .

**if**  $WCONF_{maxindex} - WCONF_{secondindex} \geq mindiff$  **then**  
 ordne  $I$  der Klasse  $maxindex$  zu, ansonsten vereinbaren wir,  
 dass  $I$  nicht klassifizierbar ist, was z.B. immer der Fall ist, wenn  
 kein Itemset der Regelmenge in  $I$  enthalten ist.  
**end**

4. Die prozentualen Angaben der Klassifikationsleistung können aus den zugeordneten Klassen im Vergleich zu den vorgegebenen Klassen berechnet werden (auf den Testdaten).

Sind alle Itemsets disjunkt und gehören genau einer Klasse an, so ist klar, dass alle Testdaten richtig klassifiziert werden bzw. nicht klassifiziert werden (wenn die Itemsets nicht häufig genug in der Trainingsmenge vorkamen, so dass in diesem Fall keine Regel existiert).

## 4.4 Ergebnisse auf Benchmark-Daten

Im Folgenden werden die Ergebnisse von Versuchen auf drei verschiedenen Datensätzen wiedergegeben, um die Funktionsfähigkeit des Verfahrens zu belegen.

**Vote-Datensatz** (symbolische Daten): Die Testmenge besteht aus 50% der Daten. Die minimale  $K_l$ -Konfidenz wurde gemäß der Testdatenaufteilung gesetzt, d.h. 0.60 für die Klasse 1 (democrat) und 0.40 für die Klasse 2 (republican);  $\min_{freq_{K_1}} := 0.044$ ,  $\min_{freq_{K_2}} := 0.034$ . Trainiert wurde (im ersten Level) mit lediglich 8% der Daten und im zweiten Level mit 50% der erzeugten Regeln aus dem ersten Level. Verwendet wurden auch die Negationen der binären Variablen. Es wurden 18 gültige Regeln erzeugt, d.h. Regeln, die die Schwellwerte erfüllen; die maximale Anzahl von Items in einer dieser Regeln betrug 3 (im Mittel: 1.3). In einem Versuch wurden 94.04% der Daten korrekt klassifiziert; nicht klassifizierbar waren 0.46% der Daten. Die Spezifität (Klasse 1 richtig erkannt) betrug 94.53%; die Sensitivität (Klasse 2 richtig erkannt) betrug 94.38%. Die mittlere Konfidenz auf den Testdaten betrug 91.24% (Kl. 1) und 61.77% (Kl. 2), die mittlere gewichtete Konfidenz 86.54% (Kl. 1) und 73.45% (Kl. 2) und die mittlere Häufigkeit 53.13% (Kl. 1) und 48.05% (Kl. 2). Die Daten des Vote-Datensatzes sind zusammenfassend gesagt gut generalisierbar.

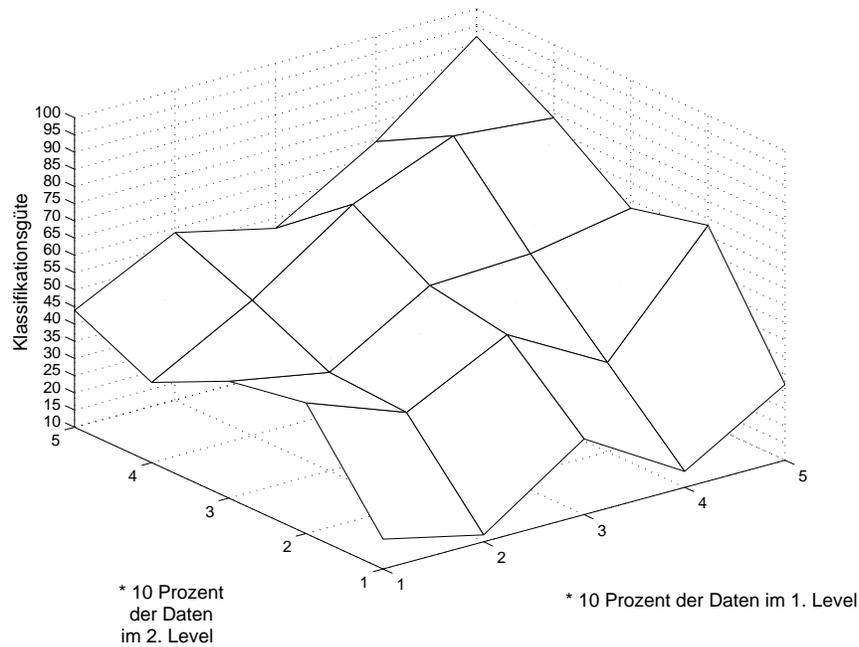
Im Vergleich hierzu erbrachte der Entscheidungsbaum C4.5 korrekte Klassifikationen von 97.3% bei 25 Regeln und (nach Pruning) 93.14% bei 7 Regeln (siehe referierte Web-Seite im Abschnitt A.1). Je nach Regelanzahl schwanken die Ergebnisse, so dass keine absolute Vergleichbarkeit besteht und man auf diesem speziellen Datensatz von einer ähnlichen Leistung ausgehen muss.

Auf die Dokumentation der durchgeführten Wiederholungen wurde verzichtet, da sie prinzipiell ähnliche Ergebnisse erbrachten und hier grundsätzlich die Möglichkeiten des Algorithmus im Mittelpunkt stehen; das gleiche gilt für die Breast-Daten. Bei der Analyse der ASK-Daten im Kapitel 6 und der symbolischen Klinikdaten im Abschnitt 7.3 werden Versuchswiederholungen dokumentiert.

**Breast-Daten** (symbolische Daten): Die Test- und Trainingsmenge besteht aus jeweils 50% der Daten. Die minimale  $K_l$ -Konfidenz wurde gemäß der Testdatenaufteilung gesetzt, d.h. 0.68 für die Klasse 1 (gutartig, engl.: benign) und 0.32 für die Klasse 2 (bösaartig, engl.: malignant);  $\min_{freq_{K_1}} := 0.55$ ,  $\min_{freq_{K_2}} := 0.13$ . Verwendet wurden auch die Negationen der binären Variablen. Es wurden 47 gültige Regeln erzeugt; die maximale Anzahl von Items in einer dieser Regeln betrug 7, die minimale Anzahl 3 (im Mittel: 5.2). In einem Versuch wurden 93.06% der Daten korrekt klassifiziert; nicht klassifizierbar waren 1.45% der Daten. Die Spezifität (Klasse 1 richtig erkannt) betrug 96.33%; die Sensitivität (Klasse 2 richtig erkannt) betrug 91.06%.

Das Ergebnis ist ähnlich zu dem einer durchgeführten logistischen Regression und auch ähnlich zu dem eines Medizinexperten (siehe referierte Web-Seite im Abschnitt A.1). Die mittlere Konfidenz auf den Testdaten beträgt 97.63% (Kl. 1) und 95.82% (Kl. 2), die mittlere gewichtete Konfidenz 95.07% (Kl. 1) und 97.98% (Kl. 2) und die mittlere Häufigkeit 58.77% (Kl. 1) und 17.99% (Kl. 2). Die Daten des Breast-Datensatzes sind damit auch sehr gut generalisierbar.

**IRIS-Datensatz** (metrische Daten): Da es bei den Daten des IRIS-Datensatzes um numerische Werte handelt, wurde hier jede der Dimensionen in 6 gleich große Abschnitte geteilt (vgl. Bem. 4.2.11). Es wurden für das 1. und 2. Level jeweils 10%, ..., 50% der Itemsets verwendet (vgl. die Bemerkung über Pruning weiter vorne). Die minimale  $K_l$ -Konfidenz wurde gemäß der Testdatenaufteilung gesetzt, d.h. in etwa jeweils 0.33;  $\min_{freq_{K_l}} := 0.05$ ,  $l = 1, 2, 3$ . Die Klassifikationsgüte kann die Leistungsfähigkeit anderer Klassifikatoren auf diesem Datensatz erreichen, hier allerdings erst bei Verwendung von 50% der Daten, siehe Abb. 4.3. Bei diesem sehr kleinen Datensatz macht das heuristische Pruning wenig Sinn. Es kann sich trotzdem lohnen, nur einen Teil der Itemsets zu betrachten in der Hoffnung auf suboptimale Ergebnisse, insbesondere bei Datensätzen mit deutlich mehr Messungen. Die Klassifikationsgüte wurde anhand der 50% Testdaten bestimmt, die im 1. Level nicht zur Regelerzeugung verwendet wurden. Da es sich jeweils um zufällige Einteilungen in Trainings- und Testdaten handelt, würde die Fläche glatter, wenn man die Mittelwerte von mehreren Versuchen aufträgt; allerdings genügt die Abbildung, um sich einen Eindruck der Leistungsfähigkeit zu machen. - Empfehlenswert ist die Regelgenerierung bei numerischen Daten dennoch nicht, da i. Allg. zu viele Regeln erzeugt werden.



**Abbildung 4.3:** 25 Klassifikationsergebnisse auf dem IRIS-Datensatz bei heuristischem Pruning von jeweils 10% bis 50% im 1. bzw. 2. Level.

Da bei den Vorstudien datendaten keine symbolischen Werte zur Verfügung standen, wird das Verfahren erst im 6. Kapitel auf symbolische Klinikdaten einer Klinik, die uns frühzeitig ihre Daten elektronisch zur Verfügung gestellt hat, angewendet, um die Eignung auf realen medizinischen Daten zu überprüfen. Dort werden die Negationen mitverwendet.

Im anschließenden Kapitel 7 werden dann die symbolischen MEDAN-Multizentren daten untersucht. Für diese Daten ist das Verfahren gut geeignet, da nur wenige Items aus einer großen Menge von Items pro Patient auftreten (z.B. hat ein Patient evtl. drei Operationen aus einer möglichen Menge von vielen hundert Operationen); es können dann die Negationen nicht mitverwendet werden, da es zu viele wären. Eine möglichst geringe Kontextverwischung (siehe Abschnitt 4.8) der Regeln ist dabei eine wichtige Eigenschaft. Ausschließen kann man sie aber aufgrund der verwendeten Heuristiken nicht gänzlich.

## 4.5 Die Maße „Wichtigkeit“ und „Frühzeitigkeit“

Die im Abschnitt 3.5 für die metrischen Regeln eingeführten Maße werden nun für die symbolischen Regeln analog definiert.

Nr.	Item	democrat	republican
1	not handicapped infants	0.129	0.856
2	not adoption of the budget resolution	0.011	0.731
3	adoption of the budget resolution	0.321	0.004
4	not physician fee freeze	<b>1.632</b>	0.000
5	el salvador aid	0.019	0.603
6	religious groups in schools	0.206	0.844
7	anti satellite test ban	0.916	0.045
8	not mx missile	0.019	0.603
9	not immigration	0.237	0.255
10	not synfuels corporation cutback	0.180	0.857
11	not education spending	1.377	0.022
12	crime	0.113	<b>0.905</b>
13	not duty free exports	0.133	0.852
14	export administration act south africa	0.354	0.244

**Tabelle 4.1:** Vote-Daten: Wichtigkeit derjenigen Items, für die Klassen „democrat“ und „republican“, die in den 18 erzeugten Regeln vorkommen, d.h. deren Wichtigkeit bez. dieser Regelmenge für eine Klasse größer 0 ist. Der höchste Wert pro Klasse ist fett markiert.

#### 4.5.1 Definition: (Wichtigkeit)

Seien  $I_1 \Rightarrow K_l, \dots, I_{r_l} \Rightarrow K_l$  die erzeugten Generalisierungsregeln der Klasse  $K_l$ , die eine Regelmenge  $\mathcal{R}$  bilden. Die **Wichtigkeit bez. der Klasse  $K_l$**  (engl.: Importance) eines Items  $A$  wird definiert als:

$$imp_{K_l}(A) := \sum_{i=1, A \in I_i}^{r_l} freq_{K_l}(I_i \Rightarrow K_l) \cdot wconf_{K_l}(I_i \Rightarrow K_l) \cdot \frac{1}{|I_i|} . \quad (4.5)$$

Die Wichtigkeit wird für den Vote-Datensatz aus dem Abschnitt 4.4 berechnet, siehe die Tab. 4.1. Man sieht in der Tab. 4.1, dass die Items 4, 7 und 11 besonders wichtig für die demokratischen und die Items 1, 2, 5, 10, 12 und 13 eher für die republikanischen Wähler waren, die Items 9 und 14 ergeben kein eindeutiges Bild. Die anderen 18 der insgesamt 32 Items (16 Items JA, 16 Items NEIN von 16 binären Variablen) können mangels jeglicher Wichtigkeit weggelassen werden.

Auch für die Items des Breast-Datensatzes wird die Wichtigkeit berechnet, siehe die Tab. 4.2. Man sieht in der Tab. 4.2, dass die Items 2, 4, 8, 10, 12 und 14 eher wichtig für die Klasse „gutartig“ und die Items 1, 3, 5, 6, 7, 9, 11, 13 und 15 eher wichtig für die Klasse „bösaartig“ sind. Die anderen 5 der insgesamt 20 Items (10 Items JA, 10 Items NEIN von 10 binären Variablen) können hier mangels jeglicher Wichtigkeit weggelassen werden.

Nr.	Item	benign	malignant
1	cellular dyshesion	0.836	1.233
2	intracytoplasmic lumina	1.584	0.368
3	not intracytoplasmic lumina	0.000	0.125
4	„3-dim.“ of epithelial cells clusters	1.198	0.004
5	not „3-dim.“ of epithelial cells clusters	0.000	0.266
6	bipolar „naked“ nuclei	0.002	2.394
7	foamy macrophages	0.000	0.076
8	nucleoli	0.901	0.004
9	not nucleoli	0.000	2.097
10	nuclear pleiomorphism	1.470	0.005
11	not nuclear pleiomorphism	0.001	2.081
12	nuclear size	1.470	0.005
13	not nuclear size	0.003	4.172
14	necrotic epithelial cells	<b>1.857</b>	1.377
15	apocrine change	0.287	<b>4.411</b>

**Tabelle 4.2:** Breast-Daten: Wichtigkeit derjenigen Items, für die Klassen „benign“ und „malignant“, die in den 47 erzeugten Regeln vorkommen, d.h. deren Wichtigkeit bez. dieser Regelmenge für eine Klasse größer 0 ist. Der höchste Wert pro Klasse ist fett markiert.

An dieser Stelle wird noch kurz die Diskussion im Anschluss an die Tab. 3.21 über die Korreliertheit der Wichtigkeitswerte fortgeführt. Während sich für das Fuzzy-RecBF-DDA hohe positive, signifikante Korrelationswerte bei 2-Klassen-Problemen ergeben haben, ergibt sich in der Tab. 4.1 ein anderes Bild: Die Werte sind hoch negativ, signifikant korreliert (Wert:  $-0.733$ ; Signifikanzniveau: 0.01). Das liegt daran, dass hier die Regeln tatsächlich anhand ihrer Häufigkeit und Konfidenz pro Klasse ausgewählt wurden, so dass hier die Wichtigkeit eines Items tatsächlich pro Klasse zu deuten ist, auch bei zwei Klassen. Ist die Wichtigkeit für eine Klasse hoch, so ist sie i. Allg. für die andere(n) Klasse(n) niedrig.

#### 4.5.2 Definition: (Frühzeitigkeit)

Ausgangspunkt seien  $s$  Zeitreihen (z.B. von Variablenmessungen eines Patienten)  $Z_j$  der Länge  $t_j$ ; jede Zeitreihe  $Z_j$ ,  $j = 1, \dots, s$  gehöre zu einer Variablen  $V_j$  mit den Ausprägungen (= Items) JA und NEIN, d.h. die Zeitreihe besteht aus binären Eintragungen. Das Itemset  $I$  besteht nun aus allen Variablennummern (Items)  $j$ , bei denen mind. einmal ein JA aufgetreten ist, d.h.  $I = \{j | \text{JA} \in Z_j\}$ . Dieses Itemset habe eine Klasse  $K_l$ . Mehrere (aus Zeitreihen von unterschiedlichen Patienten entstandene) Itemsets bilden eine Menge  $\mathcal{P}$  von Regeln. Die Regelmenge der aus  $\mathcal{P}$  generalisierten Regeln sei  $\mathcal{R}$ . Die (mittlere) **Frühzeitigkeit** einer Regel  $I \Rightarrow K_l \in \mathcal{R}$

(engl.: Earliness) sei (mit  $\text{mean}$  als arithmetischen Mittelwertoperator) definiert als:

$$\text{earliness}(I \Rightarrow K_l) := \text{mean}_{\{J \in \mathcal{P} \mid I \subset J\}} \left( \min\{t_j - t_{Z_j^{(I)}} \mid j \in I\} \right) . \quad (4.6)$$

Dabei sei  $t_{Z_j^{(I)}}$  derjenige Zeitpunkt, an dem das Item JA in der zu  $I \subset J$  gehörigen Zeitreihe  $Z_j$  zum *erstenmal* auftritt.

Die Wichtigkeit und Frühzeitigkeit wird für symbolische MEDAN-Daten in den Kapiteln 6 und 7 berechnet werden.

## 4.6 Auswahlkriterien

Wie bereits festgestellt, ist es nur bei entsprechender Rechenzeit oder bei kleinerer Problemgröße möglich, die Generalisierung mittels Durchschnitten vollständig durchzuführen. Daher ist es zweckmäßig, den Generalisierungsalgorithmus geeignet so zu modifizieren, dass er heuristisch funktioniert. Dazu werden im Folgenden einige Möglichkeiten aufgezählt und anschließend diskutiert.

1. Durchschnittsbildung nur von Itemsets, für die jeweils dieselbe Klassenzugehörigkeit für mind. eine Klasse größer 0 (bzw. größer als eine vorgegebene Schranke  $\alpha$ ) ist.
2. Begrenzung der maximalen Levelzahl.
3. Sampling, d.h. zufällige Auswahl von Regeln in jedem Level für die Durchschnittsbildung.
4. Maximale Anzahl von nichttrivialen Durchschnitten, die eine Regel in einem Level erzeugen darf.
5. Verschmelzung der Itemsets in einer zufälligen Hierarchie.
6. Beschränkung des Suchraums durch einen Generalisierungsgraphen.
7. Sequentielle Generierung von Itemsets ohne Level.

Zu 1.) Durch die disjunkte Regelbildung, getrennt für jede Klasse ( $\alpha = 0$ ) kann man bei (entsprechender Implementierung), insbesondere bei vielen Klassen, Vergleichsoperationen sparen. Einen kleinen Schönheitsfehler enthält diese Vorgehensweise, von dem aber zu vermuten ist, dass er praktisch nicht von Bedeutung ist. Es seien wieder die Itemsets  $ABCD(0, 2)$ ,  $ABDE(5, 5)$  und  $CDEF(5, 0)$  gegeben mit den ursprünglichen gezählten Häufigkeiten der Itemsets pro Klasse in Klammern dahinter. Im ersten Level würden mit der Strategie aus Punkt 1. erzeugt:  $BD(5, 7)$  und  $DE(10, 5)$ . Nicht mehr erzeugt würde  $CD(5, 2)$ , obwohl das Itemset  $CD$  von allen Itemsets dieses Levels die höchste Konfidenz für die Klasse 1 von 0.71 besitzt

( $conf_{K_1} = 0.42$ ,  $conf_{K_2} = 0.67$ ). Wählt man  $\alpha > 0$ , so wird dieser Effekt stärker in Erscheinung treten bei gleichzeitiger Erzeugung von weniger Itemsets.

Zu 2.) Die Begrenzung der maximalen Levelzahl ist nach Vorversuchen empirisch möglich und bei einer nicht sehr großen Rechnerleistung und schwer zu generalisierenden Itemsets notwendig. Sie wird empirisch bestimmt.

Zu 3.) Für das 1. Level bietet sich generell die Auswahl von weniger als 50% der Itemsets an, um die anderen Daten als Testdaten zu verwenden. Die Experimente mit dem Vote-Datensatz zeigen, dass bereits die Verwendung von deutlich weniger als 50% der Daten zu guten Generalisierungen führen kann. Insbesondere bei sehr großen Datenbanken erscheinen Vorversuche mit einem Teil der Itemsets sinnvoll. Ist die Klassifikationsperformanz dann nicht zufriedenstellend und genügend Rechnerkapazität vorhanden, dann kann man den Versuch mit mehr Itemsets wiederholen.

Zu 4.) Da bei schwer zu generalisierenden Datensätzen evtl. sehr viele Itemsets entstehen können, man heuristisch aber jedem Itemset eine Chance zur Generalisierung geben möchte, bietet es sich an, für jedes Itemset  $I$  beim Durchlauf der Itemsets  $J$  nur eine maximale Anzahl der neu erzeugten Durchschnitte  $I \cap J$  zuzulassen. Da  $I$  auf der linken und auf der rechten Seite des Durchschnittsoperators stehen kann, wurde ad hoc vereinbart, dass die maximale Anzahl, im Folgenden mit  $\max_{new}$  bezeichnet, für das Auftreten des Items  $I$  auf der linken Seite steht (bei einem Schleifendurchlauf), so dass *tatsächlich* ein Itemset maximal die doppelte Anzahl an Itemsets als  $\max_{new}$  erzeugen kann.

zu 5.) Da eine hierarchische Verschmelzung von Itemsets das Mitführen einer Graphen- oder Baumstruktur erfordert und vermutlich eine ähnliche Heuristik wie 4. bewirkt, wird aufgrund des zusätzlichen Aufwands auf die Anwendung dieser Form der Heuristik verzichtet.

Zu 6.) Ein spezieller Generalisierungsgraph („Konzepthierarchie“) benötigt spezielles Kontextwissen über die erlaubten Generalisierungen [HCC92]. Dieser Ansatz wird hier nicht verfolgt.

Zu 7.) Eine Idee besteht darin, anhand von sequentiellen Durchläufen durch die Itemsets nach und nach eine Regelmenge zu erzeugen. Sind noch keine Regeln in der Regelmenge vorhanden mit der ein neues Itemset abgedeckt werden kann, so wird eine neue Regel eingefügt. Für diesen Mechanismus benötigt man eine Kennzahl für die Ähnlichkeit von Regeln. Da dieses Vorgehen ein separates Verfahren darstellt, wird in dieser Arbeit nicht darauf eingegangen. Es könnten aber dafür die Ähnlichkeitsmaße aus dem Abschnitt 4.9 verwendet werden.

Da die Heuristiken 1.-4. eine einfache, aber wirkungsvolle heuristische Vorgehensweise ermöglichen, werden diese bei den Analysen angewendet. Die Heuristiken 5.-7.

haben bereits den Charakter eigenständiger Verfahren, so dass sie hier nicht weiter verfolgt werden. Um die Effekte dieser Heuristiken im Detail zu erforschen, wären sehr umfangreiche Versuche auf zahlreichen Testdaten nötig. Da hier aber die medizinische Anwendung im Vordergrund steht, wird die Heuristik 1. immer und die Heuristiken 2.-4. jeweils nach einigen Vorversuchen direkt bei der Datenanalyse angewendet.

Da die tatsächlich verwendeten Heuristiken zwar eine deutliche Verringerung der Laufzeit mit sich bringen, bleibt doch unklar, *wann* das Verfahren sinnvollerweise abgebrochen werden kann. Ein *adaptives* Verfahren sollte ein selbstständiges Abbruchkriterium besitzen. Aus diesem Grund führe ich im nächsten Abschnitt den *äußeren Generalisierungsindex* ein, um die maximal sinnvolle Anzahl von Leveln zur Laufzeit zu bestimmen. Außerdem führe ich den *inneren Generalisierungsindex* ein, mit dessen Hilfe ein Leveldurchlauf abgebrochen werden kann, so dass schneller mit dem nächsten Level fortgefahren werden kann. Diese Heuristiken können die o.g. ersetzen oder ergänzen.

## 4.7 Der innere und äußere Generalisierungsindex

Angenommen, man befindet sich in einem Leveldurchlauf. Die durchgeführten Versuche zeigen, dass am Anfang eines Levels viele neue Itemsets erzeugt werden, während am Ende deutlich weniger Itemsets erzeugt werden, so dass es sinnvoll sein kann, den Leveldurchlauf frühzeitig abzurechnen. Besonders profitabel ist die Kombination mit der maximalen Anzahl von neu erzeugten Itemsets  $max_{new}$  für ein Itemset  $I$ , da dann nicht mehr alle Itemsets für eine mögliche Durchschnittsbildung mit  $I$  durchsucht werden müssen. Die Idee ist nun einen gleitenden Mittelwert zu berechnen über der Anzahl der neu erzeugten Itemsets pro Itemset. Wird dieser gleitende Mittelwert zu klein, so kann der Leveldurchlauf abgebrochen werden.

Dazu ein typisches Beispiel: Sei  $max_{new} = 4$  und die Fenstergröße  $F = 2$ . Die Anzahl der erzeugten Itemsets sei der Reihe nach 4, 4, 4, 3, 4, 3, 2, 3, 4, 1, 2, 1, 0, 1, 0, 0, 0. Dann berechnen sich die gleitenden Mittelwerte, **innere Generalisierungsindices**  $G^{(in)}$  genannt, zu 4, 4, 3.5, 3.5, 3.5, 2.5, 2.5, 3.5, 2.5, 1.5, 1.5, 0.5, 0.5, 0.5, 0, 0. Ist jetzt  $s = 1$  eine Schwelle und  $r = 2$  die Anzahl der Werte  $G^{(in)}$ , die kleiner oder gleich der Schwelle sind, bevor abgebrochen wird, so würde nach dem 14. Itemset (13. Wert  $G^{(in)}$ ) abgebrochen werden: 4, 4, 3.5, 3.5, 3.5, 2.5, 2.5, 3.5, 2.5, 1.5, 1.5, **0.5, 0.5**, 0.5, 0, 0. Die Durchschnittsbildung für 3 Itemsets würde eingespart werden, wobei hier kein potenziell neues Itemset nicht erzeugt wurde. Wäre dies der Fall, so bestände aber im nächsten Level die Chance, dass dieses nicht erzeugte Itemset erzeugt werden würde.

Wie viele Level sind nun aber nötig? Für die heuristische Beantwortung dieser Frage führe ich den **äußeren Generalisierungsindex** ein. Dazu sei  $m$  die Anzahl der in einem Level überhaupt erzeugbaren neuen Itemsets und  $e$  sei die Anzahl der tat-

Level	neu erzeugt	möglich	$G^{(out)}$
1	16	-	-
2	60	120	0.500
3	196	1770	0.111
4	42	19306	0.002
5	0	903	0.000

**Tabelle 4.3:** Vote-Daten: Neu erzeugte, möglich zu erzeugende Itemsets und äußere Generalisierungsindices.

Level	neu erzeugt	möglich	$G^{(out)}$
1	68	-	-
2	184	198	0.929
3	467	546	0.855
4	1115	1395	0.799
5	2168	3339	0.649

**Tabelle 4.4:** ASK-Daten: Neu erzeugte, möglich zu erzeugende Itemsets und äußere Generalisierungsindices.

sächlich neu erzeugten Durchschnitte, die noch nicht vorhanden und nicht leer sind. Ist  $n$  eine Anzahl von Itemsets, dann könnten maximal  $m = \sum_{i=1}^{n-1} i$  neue Itemsets erzeugt werden. Arbeitet man mit der Schranke  $max_{new}$ , so können maximal  $m = max_{new} \cdot (n - max_{new}) + \sum_{i=1}^{max_{new}-1} i$  neue Itemsets erzeugt werden. Dann sei

$$G^{(out)} := \frac{e}{m} . \quad (4.7)$$

Fällt nun  $G^{(out)} \in [0, 1]$  unter eine vorgegebene Schwelle, dann kann das Verfahren vorzeitig abgebrochen werden. Ist der Index nahe bei 1, so sollte das Verfahren weiterlaufen.

In den Tabellen 4.3 und 4.4 sind die äußeren Generalisierungsindices für die Vote-Daten und die ASK-Daten berechnet. Während man bei den Vote-Daten bereits nach dem 3. oder 4. Level hätte abbrechen können, wäre es vielleicht bei den ASK-Daten ratsam ein weiteres oder weitere Level zu probieren, insbesondere in der Kombination mit dem inneren Generalisierungsindex.

Zählt man bei der Erzeugung der Durchschnitte in einem Level diejenigen Durchschnitte, die leer sind, und setzt diese Anzahl gleich  $k$ , so kann man  $G^{(empty)}$  (*Leer-Index*) definieren als

$$G^{(empty)} := \frac{k}{m} . \quad (4.8)$$

Ist  $G^{(empty)}$  nahe bei 1, so läßt sich die Menge der Itemsets nicht gut generalisieren in dem Sinne, dass die Itemsets sehr unterschiedlich sind. Ist  $G^{(empty)}$  nahe bei 0, so heißt das aber nicht unbedingt, dass sich die Menge der Itemsets gut generalisieren lässt. Es kann immer noch sein, dass die erzeugten Itemsets nicht leer sind, aber bereits in früheren Leveln erzeugt wurden bzw. initial bereits vorhanden waren. Sei  $v$  die Anzahl der nicht leeren, bereits vorhandenen Durchschnitte, dann definiere ich

$$G^{(exist)} := \frac{v}{m} . \quad (4.9)$$

Liegt  $G^{(exist)}$  (*Existenz-Index*) nahe bei 1, dann enthält die Menge der Itemsets bereits ihre generalisierten Itemsets.

Man kann sagen, dass sich die Indices  $G^{(empty)}$  und  $G^{(exist)}$  ergänzen und gemeinsam komplementär zum Index  $G^{(out)}$  sind, da

$$1 - G^{(out)} = G^{(empty)} + G^{(exist)} \quad (4.10)$$

gilt. Daher eignet sich  $G^{(out)}$  am ehesten als Abbruchkriterium.

## 4.8 Vergleich zu Assoziationsregeln

An dieser Stelle möchte ich ein paar grundsätzliche Überlegungen zu Generalisierungsregeln im Vergleich zu Assoziationsregeln anstellen, insbesondere mit einer Betonung auf den medizinischen Nutzen der Regeln für Ärzte. Ausgangspunkt sind dabei Mengen von Itemsets mit unscharfer Klassenzugehörigkeit.

Ein Assoziationsregelmechanismus (im Folgenden kurz: A-Alg.), der von Items ausgeht, bildet sehr viele Regeln. Angenommen, man hätte nur die beiden Regeln  $ABC$ ,  $BCD$  als initiale Regeln, die beide zu einer Klasse gehören. Dann erzeugt der Generalisierungsalgorithmus (im Folgenden kurz: G-Alg.) nur die zusätzliche Regel  $BC$ , während bei der Assoziationsregelgenerierung zunächst die Items  $A, B, C$  und  $D$  ausgewertet und dann die Kombinationen  $AB, AC, AD, BC, BD, CD$  gebildet werden.  $AD$  ist ein Itemset, das nicht in den ursprünglichen Itemsets enthalten ist. Häufig wäre auf jeden Fall das Itemset  $BC$ , aber auch  $B$  und  $C$ .

Die A-Alg. besitzen die Eigenschaft, zu viele *notwendige*, aber nicht unbedingt *hinreichende* Regeln zu erzeugen. Im Sinne von Satz 4.2.3 ist die Regel  $BC$  bereits optimal bez. der Konfidenz und der Häufigkeit bez. der Klassenzugehörigkeit.  $B$  und  $C$  liefern keine zusätzlichen, sinnvollen Informationen, falls man klassifizieren möchte.

Alle häufigen Itemsets zu erzeugen ist nicht unser Ziel und führt bei A-Alg. zu einem Effekt, den ich als *Kontextverwischung* (engl.: context smearing) bezeichnen möchte,

da sich den Regeln von den Items her genähert wird und nicht von den Itemsets, die den Kontext ausmachen. Der A-Alg. erzeugt zu viele Regeln.

Bezeichnet  $B$  z.B. einen hohen Blutdruck und  $C$  einen niedrigen pH-Wert, so stellt nur eine Generalisierung sicher, dass die *Kombination* als das entscheidende Wissen erkannt wird. Die Assoziationsregeln lassen vermuten, dass auch der hohe Blutdruck oder der niedrige pH-Wert alleine interessant sind, was zwar möglich ist, aber nicht sein muss. Würde man einem Arzt nur das Item  $B$  als generalisierte Regel präsentieren, so könnte er die falschen Schlüsse daraus ziehen: Er könnte ein Blutdrucksenkendes Medikament verabreichen, das zum Ungunsten des Patienten z.B. den pH-Wert weiter senkt, da er nicht die vollständige Information  $BC$  hatte. Auch ist die eine Regel  $BC$  einer Kombination der Art „ $B$  UND ( $B \Leftrightarrow_P C$ )“ vorzuziehen, da letztere Regel komplizierter ist und Äquivalenzen gefunden werden müssten, die mit einer Wahrscheinlichkeit  $> P$  gelten.<sup>6</sup>

Anhand der hier begründeten Kontextverwischung ist das Prinzip einer möglichst geringen Spezialisierung mit wenigen Items im Sinne eines A-Alg. eher kritisch zu sehen, da die kürzeren Beschreibungen den Kontext verwischen, obgleich sie für Datenbankabfragen, die ja auf Attributen basieren, evtl. Kosten einsparen können.

An dieser Stelle wird nun folgende Frage diskutiert: Wie sinnvoll werden mit einer Regelmenge *unbekannte* Daten klassifiziert, d.h. wie *robust* ist ein A- bzw. G-Alg.? Dazu ein kleines Beispiel: Seien wieder die beiden Itemsets  $ABC$  und  $BCD$  (derselben Klasse  $s$ ) gegeben. Die vom G-Alg. erzeugte Regelmenge sei  $BC$ , die vom A-Alg. erzeugte  $B$  bzw.  $C$ . Das unbekannte, zu klassifizierende Itemset sei  $AB$ . Es gilt  $BC \not\subset AB$ , d.h. im Falle der Regelmenge des G-Alg. würde (sinnvollerweise!)  $AB$  als unbekannt bzw. nicht klassifizierbar ausgegeben werden.  $AB$  könnte ja einer anderen Klasse als  $BC$  angehören; also ist die Ausgabe sinnvoll. Aber es gilt  $B \subset AB$ , d.h. im Falle der Regelmenge  $B$  des A-Alg. würde  $AB$  (mit einer Wahrscheinlichkeit  $> 0$ ) als zur Klasse  $s$  gehörig ausgegeben werden, obwohl  $AB$  in der Realität nicht zur Klasse  $s$  gehören muss. Wegen  $C \not\subset AB$  würde im Fall der Regelmenge  $C$  das *abweichende* Ergebnis „ $AB$  unbekannt“ ausgegeben werden. In der Tat wird im Fall  $B$  ein i. Allg. unzulässiger Schluss, eine willkürliche Zuordnung durchgeführt. D.h. der A-Alg. erzeugt nicht robuste Regelmengen, in der es teilweise vom Zufall abhängt, ob unbekannte Itemsets richtig oder falsch klassifiziert werden. Das ist eine Folgerung der Kontextverwischung des A-Alg.

Schwachpunkte eines A-Alg. sind zusammenfassend die Erzeugung nicht vorhandener Itemsets, Erzeugung vieler elementarer Kombinationen, die Voraussetzung einer (willkürlichen) Häufigkeitsschwelle, ungenügende Robustheit und insbesondere die Kontextverwischung. Positive Aspekte sind die Verwendbarkeit von effizienten attributorientierten Datenbankzugriffstechniken und die Verwendbarkeit der Monotonieeigenschaft der Häufigkeit zum Prunen von Itemsets.

---

<sup>6</sup>Exakte logische Äquivalenzen werden fast nie vorhanden sein.

G-Alg. bieten den Vorteil, dass sie den Kontext nicht verwischen und robuster sind. Bei vielen ähnlichen Itemsets kann, insbesondere bei Verwendung von Heuristiken, schnell generalisiert werden. Die Durchschnittsbildung ist abgeschlossen, d.h. es werden keine Itemsets erzeugt, die die Häufigkeit 0 haben. Es muss nicht in Stufen ein Item nach dem anderen generalisiert werden, sondern es können viele Items gleichzeitig generalisiert werden. Ein Nachteil ist die Überprüfung, ob erzeugte Durchschnitte bereits vorhanden sind, der aber durch effiziente Datenstrukturen abgemildert werden kann. Da die Generalisierungsregeln i. Allg. länger sind als Assoziationsregeln, kann die Verständlichkeit der Regeln für einen Anwender etwas schwieriger werden, dafür bekommt der Anwender aber auch die vollständigen, hinreichenden Informationen, die die Gefahr einer Fehlinterpretation der Regeln vermeiden helfen.

Hat man eine Regelmenge erzeugt, sei es mit einem A- oder G-Alg., so kann der Wunsch des Anwenders auftreten, aus einer präsentierten Regelmenge eine kleinere Anzahl von sehr unterschiedlichen Regeln („markante Regeln“) einzusehen, um einen Überblick über die verschiedenen erzeugten Regeln zu bekommen. Die Hypothese hierzu lautet dann, dass eine Regelmenge interessant ist, wenn die Regeln aus der Menge möglichst unähnlich sind, d.h. dem Anwender keine Redundanz präsentiert wird. Im nächsten Abschnitt werde ich eine Kenngröße entwickeln, die es erlaubt, solch eine Untermenge von Regeln aus einer vorgegebenen Regelmenge zu identifizieren.

## 4.9 Ähnlichkeit von Generalisierungsregeln

Es kann z.B. sein, dass die Regeln  $I = ABCDE$  und  $J = BCDEF$  für die gleiche Klasse erzeugt wurden, allerdings mit nicht identischen Häufigkeits- und Konfidenzwerten. Sind die Itemsets  $I$  und  $J$  ähnlich? Eine einfache Möglichkeit eine Kenngröße  $sim$  für die Ähnlichkeit zu definieren ist:

$$sim_1(I, J) := \frac{|I \cap J|}{\max\{|I|, |J|\}} . \quad (4.11)$$

Dann wäre im Beispiel  $sim_1(I, J) = 4/5 = 0.8$ . Die Itemsets würden als ähnlich eingestuft werden. Ist das sinnvoll? Angenommen  $\mathcal{K}$  sei die Menge aller für den Anwender interessanten Kenngrößen, also z.B.  $\mathcal{K} = \{freq, conf\}$ . Diese Kenngrößen seien vereinheitlichend mit  $G_1, G_2, \dots$  bezeichnet. Z.B. könnte es sein, dass im Beispiel trotz der mengenbasierten Ähnlichkeit von  $I$  und  $J$  mittels  $sim_1$  die Häufigkeit und die Konfidenz von  $I$  und  $J$  sich deutlich unterscheiden, z.B.  $freq(I) = 0.2$ ,  $freq(J) = 0.7$ ,  $conf(I) = 0.9$  und  $conf(J) = 0.6$ . Diese wichtigen Unterschiede würden in der Kenngröße (4.11) nicht eingehen. Um diese Unterschiede zu berücksichtigen, schlage ich vor, die Kenngröße für die **Ähnlichkeit** (4.11) um folgende Terme pro Kenngröße  $G_i$  zu ergänzen:

$$1 - |G_i(I) - G_i(J)| . \quad (4.12)$$

Vorausgesetzt ist hier, dass die Kenngrößen zwischen 0 und 1 normiert sind wie die Häufigkeit oder die Konfidenz. Man erhält dann mit  $g \in \mathbb{N}$  Termen der Art (4.12) aus (4.11):

$$\text{sim}_2(I, J) := \frac{\text{sim}_1(I, J) + \sum_{i=1}^g \kappa_i (1 - |G_i(I) - G_i(J)|)}{1 + \sum_{i=1}^g \kappa_i} . \quad (4.13)$$

Wählt man die Gewichtung  $\kappa_i \in [0, 1]$  nahe bei 0, so ist der Einfluss der Kenngröße klein, wählt man  $\kappa_i$  nahe bei 1, so ist der Einfluss der Kenngröße groß.

Der Vorteil von (4.13) liegt in der Berücksichtigung aller für eine Anwendung wichtigen Kenngrößen. Im Beispiel ergibt sich mit  $\kappa_1 = \kappa_2 = 0.5$ :  $\text{sim}_2(I, J) = (4/5 + 0.5(1 - (0.7 - 0.2)) + 0.5(1 - (0.9 - 0.6)))/(1 + 0.5 + 0.5) = (0.8 + 0.25 + 0.35)/2 = 0.7$ . Wäre  $\text{freq}(I) = 0.5$ ,  $\text{freq}(J) = 0.6$ ,  $\text{conf}(I) = 0.9$  und  $\text{conf}(J) = 0.8$ , dann wäre  $\text{sim}_2(I, J) = (4/5 + 0.5(1 - (0.6 - 0.5)) + 0.5(1 - (0.9 - 0.8)))/(1 + 0.5 + 0.5) = (0.8 + 0.45 + 0.45)/2 = 0.85 > 0.7$  und die Itemsets wären ähnlicher.  $\text{sim}_2$  berücksichtigt also die relevanten Kenngrößen angemessen und geht nicht nur von der Mengendarstellung aus.

Mit Hilfe des Ähnlichkeitsmaßes kann eine Clusteranalyse für die Regeln aus der Regelmenge durchgeführt werden. Dann ist es möglich die Clusterzentren als „markanteste Regeln“ auszugeben.

Ein anderes Problem, bei dem die hier eingeführte Ähnlichkeit von Nutzen ist, ist die Auswahl einer Regelmenge aus einer Menge von Regelmengen, die durch zufällige Versuchswiederholungen entstanden sind. Diese „Meta-Modellwahl“ wäre manuell nicht möglich.

Seien  $R = \{I_1, \dots, I_m\}$  und  $\hat{R} = \{J_1, \dots, J_n\}$  zwei Regelmengen und  $S = (s_{jk}) := (\text{sim}_2(I_j, J_k))$  die Matrix aller Ähnlichkeiten zwischen Regeln der Regelmenge  $R$  und  $\hat{R}$ . Ein Ähnlichkeitsmaß zwischen Regelmengen sollte vergleichen, wie ähnlich Regeln der einen Regelmenge zu denen der anderen sind. Allerdings ist klar, dass eine Regel aus  $R$  nicht zu allen anderen Regeln aus  $\hat{R}$  ähnlich sein muss, sondern zu mindestens einer Regel aus  $\hat{R}$  ähnlich sein sollte. Außerdem sollte die Ähnlichkeit symmetrisch sein. Das motiviert die folgende Definition der **Ähnlichkeit von Regelmengen**  $\text{sim}(R, \hat{R})$ , bei der zusätzlich noch die Anzahl der Regeln in den betrachteten Regelmengen mit einem Gewichtungsfaktor berücksichtigt wird. Dabei sei  $m = \#R$  und  $n = \#\hat{R}$ .

$$\text{sim}(R, \hat{R}) := \frac{\frac{\sum_{j=1}^m \max_k \{s_{jk}\} + \sum_{k=1}^n \max_j \{s_{jk}\}}{m+n} + \tau \left(1 - \frac{\min\{m, n\}}{\max\{m, n\}}\right)}{1 + \tau} . \quad (4.14)$$

Berechnet man die Ähnlichkeit zwischen mehreren Regelmengen, so können wegen der Symmetrie von  $\text{sim}(R, \hat{R}) \in [0, 1]$  die Werte als obere Dreiecksmatrix gespeichert werden. Die Diagonale mit Eins-Einträgen muss nicht mit abgespeichert werden.

Seien nun  $r$  Regelmengen  $R_1, \dots, R_r$  gegeben. Dann ist diejenige Regelmenge  $R_j$  **repräsentativ**, für die gilt:  $j = \text{index}(\max_{i=1, \dots, r} \{Sim_i\})$  mit

$$Sim(R_i) := \frac{\sum_{j=1, i \neq j}^r sim(R_i, R_j)}{r - 1} . \quad (4.15)$$

Die repräsentative Regelmenge ist also diejenige, die im Mittel am ähnlichsten zu allen anderen ist. Sie ist damit statistisch ausgezeichnet.

Bei der Auswahl einer Regelmenge aus  $r$  Regelmengen ist die repräsentative die für eine Anwendung geeignetste. Klar ist, dass die Berechnung einer repräsentativen Regelmenge erst bei mindestens 3 Mengen möglich ist.

Im Abschnitt 3.7 wurde die Ähnlichkeit von Regeln und Regelmengen für die metrischen Regeln des Fuzzy-RecBF-DDA definiert, und es werden Versuchsergebnisse zu diesen Definitionen präsentiert. Im Abschnitt 6.4 werden die Regeln der ASK-Daten gemäß ihrer Ähnlichkeit geclustert.

Ergänzung: Eine grundsätzlich andere Möglichkeit zu einem Modell aus einer Vielzahl von Modellen zu gelangen, ist die *Kombination* der einzelnen Modelle, siehe [Hay99, Kap. 7]. Man erhält dadurch ein *komplexeres* Modell (eine sog. Komitee-Maschine, engl.: committee machine) mit einer verbesserten Klassifikationsleistung. Eine Regelinterpretation wird durch die Betrachtung komplexer, zusammengesetzter Modelle allerdings erschwert.

Im sog. *case-based reasoning* (Abk.: CBR, dt.: Fall-basiertes Schließen, [BKI00, Kap. 6]) werden anhand von Ähnlichkeitsmaßen repräsentative Fälle z.B. in einer Datenbank bestimmt, um Klassifikationen von unbekanntem Daten (=Fällen) vorzunehmen. Eine gewisse Analogie besteht zum o.g. Vorgehen, insofern man hier die Regelmengen als Fälle interpretieren kann. Dann wird eine repräsentative Regelmenge aufgrund von Ähnlichkeiten selektiert. Allerdings dient die repräsentative Regelmenge hier nicht als Vergleich für unbekanntem Regelmengen, sondern sie wird als repräsentatives Analyseinstrument verwendet.

# Kapitel 5

## Die metrischen septischer Schock-Patientendaten der Vorstudie

Dem Anliegen, Erkenntnisse über den septischen Schock zu gewinnen, wird in diesem und den nächsten beiden Kapiteln verstärkt Rechnung getragen. Als Erstes wird eine Einordnung der Begriffe *SIRS*, *Sepsis* und *septischer Schock* gegeben sowie die prinzipielle Aufgabenstellung samt ihrer Bedeutung für die Medizin dargelegt. Danach werden im Abschnitt 5.2 die notwendigen Vorverarbeitungen geschildert. Im Abschnitt 5.3 werden die Ergebnisse von Korrelationsanalysen vorgestellt.

Anschließend werden die Ergebnisse der Zustandsanalyse und der Regelgenerierung wiedergegeben (Abschnitte 5.4 und 5.5). Die *Wichtigkeit* der Variablen, die *Frühzeitigkeit* und die *Ähnlichkeit* der generierten Regeln wird diskutiert. Am Schluss wird die Regel eines Medizinexperten analysiert. Alle Untersuchungen werden in diesem Kapitel anhand der *Vorstudiendaten* durchgeführt.

### 5.1 SIRS - Sepsis - septischer Schock

Über die Definitionen des *SIRS* (engl., Abk. für: Systemic Inflammatory Response Syndrome), der *Sepsis* und des *septischen Schocks* wird in der Medizin kontrovers diskutiert. In dieser Arbeit werden die Definitionen der *Concensus Conference ACCP/SCCM* von 1991 verwendet [B<sup>+</sup>92]. Sie sind in der Arbeit [WBH98] beschrieben und werden von dort übernommen.

**5.1.1 Definition:** a) Eine **SIRS** liegt vor, wenn zwei oder mehr der folgenden Bedingungen erfüllt sind:

- Temperatur  $Temp$  [°C]:  $Temp < 36$  oder  $Temp > 38$ ,
- Herzfrequenz  $Herz$  [Schläge/min]:  $Herz > 90$ ,

- Atemfrequenz  $Atem$  [Atemzüge/min]:  $Atem > 20$   
oder  $PaCO_2$  [mmHg]:  $PaCO_2 < 32$ ,
- Anzahl der Leukozyten  $Leuko$  [/mm<sup>3</sup>]:  $Leuko < 4000$  oder  $Leuko > 12000$ .

b) Unter einer **Infektion** versteht man gemäß [Psc94] die Übertragung, das Haftbleiben und das Eindringen von Mikroorganismen (Viren, Bakterien, Pilze, Protozoen u.a.) in einen Mikroorganismus (Pflanze, Tier, Mensch) und Vermehrung in ihm.

c) Eine **Sepsis** liegt vor, wenn gilt:

- SIRS kann nachgewiesen werden und
- eine Infektion kann nachgewiesen werden.

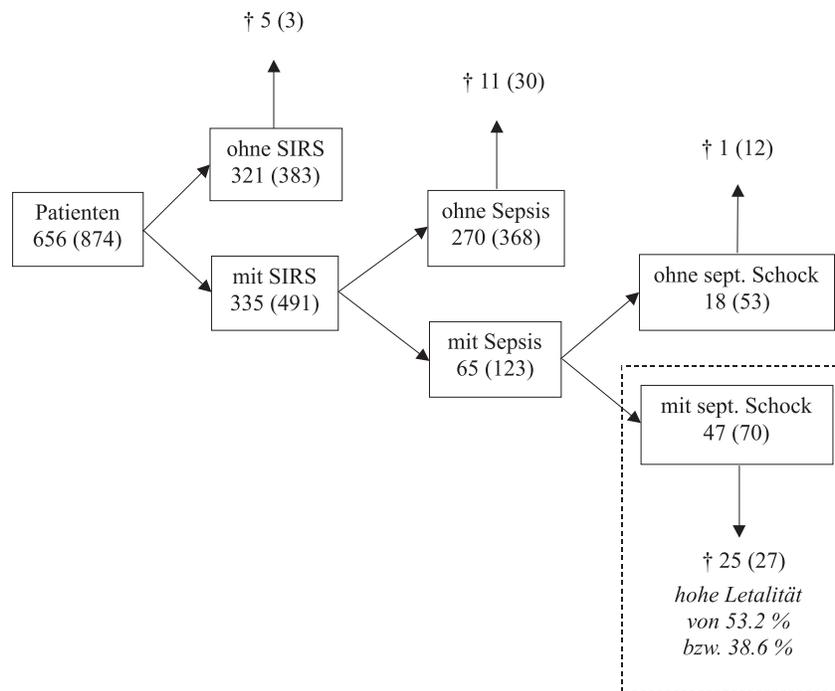
d) Patienten befinden sich im Zustand des **septischen Schocks**, wenn sie die Bedingungen der Sepsis erfüllen und trotz adäquater Volumensubstitution hypoton sind, d.h. sie haben einen niedrigen systolischen Blutdruck  $< 90$  mmHg. Erhalten die Patienten vasokonstriktive, d.h. gefäßverengende Substanzen (Katecholamine), kann die Phase der Hypotension nicht exakt ermittelt werden; die Patienten werden hier als im septischen Schock betrachtet.

Die medizinische Problematik des septischen Schocks (hohe Letalität) wird in [HE01], [SWR98] und [Har00] hervorgehoben. In [F<sup>+</sup>97], [NRRS98] wird verstärkt auch auf sog. *Mediatoren* eingegangen. Mediatoren sind laut [Psc94] hormonähnliche Wirkstoffe, die von verschiedenen Geweben bzw. Zellen des Organismus produziert werden und auf charakteristische Weise wirksam werden. Auf die Bedeutung dieser Stoffe soll im Folgenden aber nicht weiter eingegangen werden, da sie im Rahmen der MEDAN-Analysen nicht zur Verfügung standen. Der Vermutung, dass immunologische Mechanismen für den septischen Schock verantwortlich sind, kann deshalb hier nicht nachgegangen werden, da auch hierüber keine (teureren und aufwendigeren) Messungen vorliegen. Verwendet werden alle üblicherweise auf der Intensivstation erhobenen Daten.

In Kürze werden die Ergebnisse der Studie aus [WBH98] wiedergegeben, anhand derer die Bedeutung des *septischen Schocks* deutlich wird. In einer zweiten Studie wurden weitere Daten gesammelt und denen der Studie [WBH98] hinzugefügt. Die Gesamtergebnisse sind in Klammern beigefügt. Der gesamte Datensatz wird als *Vorstudie* bezeichnet, vgl. den Anhang A.<sup>1</sup>

---

<sup>1</sup>Die Patienten der Vorstudie wurden *so gut wie möglich* von Medizinern nach oben genannten Kriterien eingestuft. Unter Medizinern herrscht an dieser Stelle aber keine Einigkeit. Die Bewertung in [WBH98] unterscheidet sich in der Bewertung der gesamten Vorstudienaten, die von einem anderen Mediziner durchgeführt wurde. Daher entstehen die unterschiedlichen Zahlenangaben. Eine *automatische* Auswertung der Kriterien bez. der Vorstudie war nicht vollständig möglich, da einige Variablen nicht mit der notwendigen Genauigkeit gemessen wurden.



**Abbildung 5.1:** Anzahl der Patienten aus der Studie [WBH98] und der gesamten Vorstudie in Klammern.

- Dauer der 1. Studie: November 1993 bis November 1995 (und 2. Studie: November 1993 bis Dezember 1997),
- Patientenzahl: 656 (874),
- Patienten ohne SIRS: 321 (383), davon verstorben: 5 = 1.6% (3 = 0.8%),
- Patienten mit SIRS: 335 (491), davon verstorben: 37 (66),
- Patienten mit SIRS, aber ohne Sepsis: 270 (368), davon verstorben: 11 = 4.1% (30 = 8.1%),
- Patienten mit SIRS und Infektion, also mit Sepsis: 65 (123),
- Patienten mit Sepsis, aber ohne septischen Schock: 18 (53), davon verstorben: 1 = 5.6% (12 = 22.6%),
- Patienten mit septischem Schock: 47 (70), davon verstorben: 25 = 53.2% (27 = 38.6%).

Die Ergebnisse sind in der Abb. 5.1 illustriert.

Die hohe Letalität (53.2% bzw. 38.6%) des septischen Schocks zeigt deutlich die Notwendigkeit weiterer Untersuchungen dieser Patientengruppe. Ziel des MEDAN-Projekts ist die Senkung der Letalität des septischen Schocks mit Hilfe eines Systems, das auf adaptiven Datenanalysen beruht. Diese Arbeit dient dazu, mit Hilfe der bereits vorgestellten Analyseverfahren die Erreichbarkeit dieses Ziels zu untersuchen.

Für die Analyse des septischen Schocks kann man die zahlreichen Variablen gruppieren, z.B. in Stammdaten (Alter, Geschlecht, ...) und Organsysteme (Herz, Lunge, Leber, ...). Zusammen mit der AG Intensivmedizin wurde solch eine Einteilung in Subsysteme vorgenommen, die im Anhang A wiedergegeben ist.

Die Analysen umfassen insbesondere verschiedene Datenvorverarbeitungen, Korrelationsanalysen, Analysen zum Auffinden wichtiger Variablen, Klassifikationsanalysen zum Auffinden von Zuständen und Regelgenerierungen. Die technischen Aspekte der eingesetzten Verfahren wurden bereits verstärkt in den vorhergehenden Kapiteln behandelt. Nach der Aufarbeitung des medizinischen Wissens und der Formulierung der Problemstellung sowie der Implementierung des Datenzugriffs folgt als nächster Schritt im Datenanalyse-Prozess (vgl. die Abb. 2.1) die *Datenvorverarbeitung* der Vorstudien-daten.

## 5.2 Vorverarbeitung der Daten

In diesem Abschnitt wird auf die Struktur der medizinischen Zeitreihen eingegangen, die aufgetretenen teilweise typischen Probleme für die medizinische Datenspeicherung bzw. -analyse werden geschildert, und die Vorschläge für eine Vorverarbeitung werden genannt [PHT00].

Die Vorstudien-daten setzen sich wie bereits beschrieben aus zwei *verschiedenen* Studien zusammen. Um diese beiden Studien zu *einer gemeinsamen* zu verschmelzen, wurde eine Liste der Variablen aufgestellt. Viele Variablen waren nur in einer Studie vorhanden. Teilweise wurden unterschiedliche Einheiten verwendet. Einige Zeitstempel fehlten. Datenbankeinträge wie „siehe oben“ in numerischen Feldern oder „Null“ für *nicht gemessen* bzw. *echt gemessene Null*, die auf mangelhaftes Datenbankdesign zurückzuführen sind, konnten retrospektiv nicht sinnvoll zugeordnet werden, so dass diese Einträge verloren gingen.

Auch waren Tippfehler vorhanden, die durch Grenzwertbetrachtungen der Werte ausfindig gemacht, aber nur teilweise korrigiert werden konnten, z.B. kann der Blutdruck nicht 1200 betragen, sondern muss 120.0 lauten. Tippfehler bei Datumsangaben, z.B. 03.12.96 statt 30.12.96 konnten durch Vergleiche mit dem Einlieferungs- und Entlassungsdatum aufgedeckt werden.

Variable	16.12.96	17.12.96	18.12.96	19.12.96
Herzfrequenz	85	125	90	90
Syst. Blutdruck	95	-9999	140	140
Temperatur	38.2	37.5	-9999	37.9
PaPm	-9999	-9999	-9999	22
...	...	...	...	...

**Tabelle 5.1:** Vorstudien­daten: Schema der Daten nach einem 24h-Sampling.

Nachdem man eine Reihe von Fehlern (unter großem Zeitaufwand mit vielen Rückfragen an Ärzte und Software-Entwickler) beseitigt hat, erhält man als Rohdaten sehr *inhomogene* Daten: Jeder Patient hat einen unterschiedlich langen Aufenthalt auf der Intensivstation. Für jeden Patienten wird eine unterschiedliche Anzahl von Variablen (Messwerte, Medikamente, Diagnosen und Therapien) zu unterschiedlichen Zeitpunkten mit einer unterschiedlichen Messhäufigkeit aufgenommen. An dieser Stelle möchte ich deutlich auf die Schwere einer Analyse mit retrospektiven medizinischen Daten dieses Umfangs hinweisen; es besteht ein großer Unterschied zum Arbeitsaufwand mit vorbereiteten Benchmark-Datensätzen, die man sich via Internet kopieren kann und die gerne als Anwendungsbeispiel verwendet werden, wie z.B. auch bereits in dieser Arbeit in den Kapiteln 3 und 4 zum Testen der Verfahren.

Für die medizinischen Daten wurden Programme zur Patienten-, Variablen- und Zeitraumauswahl geschrieben.

Ergänzung: Als MATLAB-Datenstruktur für die Zeitreihendaten wurde ein Cell-Array von Matrizen gewählt: `ts{1}(3,4)` bedeutet: 3. Zeitstempelpunkt für die 4. Variable für den Patienten mit der Nr. 1; die Zeitstempel werden in einer separaten Datenstruktur mitgeführt; in `date{1}(3)` steht der Zeitstempel für den 3. Eintrag des 1. Patienten.

Um die schwierig handhabbare Zeitreihenstruktur den Analysen zugänglich zu machen, wurde für die Vorstudien­daten in Absprache mit den Medizinern ein 24h-Sampling gewählt, in der der jeweils erste gemessene Wert pro Tag verwendet wurde; ein anderes Sampling war für diese Daten nicht möglich.

Ergänzung: Eine elegante, für das Projekt aus Zeit- und Kostengründen nicht realisierbare Möglichkeit wäre der Aufbau einer *temporären* Datenbank unter Verwendung der Abfragesprache TSQL2 (siehe [ZCF<sup>+</sup>97, Kap. 5]).

Am Ende dieses Vorverarbeitungsschrittes erhält man eine Datenbank mit der in der Tab. 5.1 schematisierten Struktur. Aufgrund der unregelmäßigen Messungen und durch zusätzliche Fehler verbleiben viele *fehlende Werte* (intern kodiert mit -9999), d.h. Felder ohne Eintrag.

Variable	Häufigkeit in %
Syst. Blutdruck	83.27
Herzfrequenz	83.08
Temperatur	82.69
ZVD	82.69
Hämoglobin	82.64
Leukozyten	82.59
Natrium	82.54
Hämatokrit	82.54
pH	82.16
pCO <sub>2</sub> Arteriell	82.11
...	...
Suprarenin	4.74
Dobutrex	4.26
Natriumbikarbonatgabe	2.13
CHE	2.13
Lipase	1.45

**Tabelle 5.2:** Vorstudiendaten: Messhäufigkeit der 10 häufigsten und der 5 seltensten Messungen nach einem 24h-Sampling für die septischen Schock-Patienten der Vorstudiendaten. Die geringe Messhäufigkeit bei den *Medikamenten* ergibt sich u.a. durch die fehlenden Einträge für *nicht verabreicht*.

Durch das *medizinisch sinnvolle Auffüllen* von Daten könnte man einige dieser fehlenden Werte sinnvoll durch einen Wert ersetzen, z.B. bei einem selten gegebenen Medikament kann man -9999 auf 0 (nicht gegeben) setzen. Von den Medizinern wird diese Variante wegen ungenauer Dokumentation der Vorstudiendaten aber als bedenklich eingestuft, so dass hier darauf verzichtet wurde.

Die Häufigkeit der gemessenen Daten wird in der Tab. 5.2 und die Zeitreihendynamik von Variablen in der Tab. 5.3 wiedergegeben. Man erkennt deutliche Abweichungen von der erwünschten Messhäufigkeit, die auf eine Beschränkung der maximalen Variablenzahl für multivariate Analysen führen bzw. zur Nicht-Durchführbarkeit von Analysen für einzelne Variablen.

Nach einer grundlegenden Vorverarbeitung der Daten kann man einige Standard-Analysen durchführen. Die Hoffnung ist zwar gering für ein komplexes Problem wie das des septischen Schocks durch einfache Analysen Ergebnisse zu erzielen, andererseits wäre es ärgerlich ein leicht zu erhaltendes Ergebnis zu „übersehen“. Dazu werden diese Analysen automatisiert, so dass *alle* Daten ausgewertet werden können; der Mediziner grenzt sein Datenmaterial i.d.R. eher ein, um Analysen „per

Variable	Arzt			tatsächlich		
	d	h	min	d	h	min
Herzfrequenz	0	1	0	1	12	14
Syst. Blutdruck	0	1	0	1	12	11
KreatininKinase	1	0	0	10	22	54
pO <sub>2</sub>	0	4	0	1	17	8
pCO <sub>2</sub>	0	4	0	1	16	59
BaseExcess	0	4	0	1	18	24
Leukozyten	1	0	0	1	12	45
Laktat	0	12	0	5	0	53
Dobutrex	0	0	30	7	13	17

**Tabelle 5.3:** Vorstudiendaten: Zeitreihendynamik der Variablen: vom Arzt erwünschte bzw. erwartete Messhäufigkeit und tatsächliche Messhäufigkeit, demonstriert anhand einiger wichtiger Beispiele. Der Unterschied bei *Dobutrex* ergibt sich u.a. durch die fehlenden Einträge für *nicht verabreicht*.

Hand“ durchzuführen oder um kommerzielle Analyseprodukte anzuwenden, da er mit dieser Art von Standardanalysen vertraut ist.

### 5.3 Deskriptive Statistik

Für die Vorstudiendaten wurden Histogramme pro Variable für alle Patienten bzw. nur für septische Schock-Patienten für die Dauer des Aufenthalts in der Intensivstation erstellt, unterschieden nach überlebenden und verstorbenen Patienten. Statistische Standard-Maße (Mittelwert, Standardabweichung) wurden berechnet. Die Variablen sind i.d.R. normalverteilt (was man auch anhand von Q-Q-Plots sehen kann, vgl. aber den Abschnitt 7.1). In den verschiedenen Histogrammen sind Unterschiede zu sehen, die aber bei keiner Variablen zu einer signifikanten Trennung zwischen verstorbenen und überlebenden Patienten führten [HPT+00].

Von den durchgeführten Korrelationensberechnungen auf den fehlerbereinigten Rohdaten (ohne Sampling) waren zwei Ergebnisse interessant:

1. Seien  $X_s, X_d$  die Mengen der Werte einer gemessenen Variablen und  $Y_s, Y_d$  die Menge der Werte eines Medikaments für überlebende (engl.: survived) bzw. verstorbene (engl.: deceased) Patienten. Betrachtet wurden alle Tage von allen Patienten.<sup>2</sup> Für 27 Paare gilt:  $|\widetilde{\text{COR}}(X_s, Y_s) - \widetilde{\text{COR}}(X_d, Y_d)| > 0.3$ , d.h. die Medikamentengaben beeinflussen stark die Korrelationswerte.

<sup>2</sup>Eine Einschränkung auf einzelne Tage oder auf die Patientengruppe des septischen Schocks war nicht möglich, da dort i.d.R. zuwenige Einzelmessungen vorlagen; gefordert wurden mind. 20 Einzelmessungen mit mind. 5 von Verstorbenen und 5 von Überlebenden.

Variable X	Variable Z	$\widetilde{\text{COR}}(X_s, Z_s)$	$\widetilde{\text{COR}}(X_d, Z_d)$	Differenz
Lipase	Kreatinin	0.52	-0.21	0.73
O <sub>2</sub> -Konz. Inspir.	Lipase	0.53	-0.20	0.73
LDH	Lipase	0.42	-0.23	0.65
KOD	Gesamtprotein	0.76	0.11	0.65
Hämoglobin	CHE	0.27	-0.32	0.59
NH <sub>3</sub>	KOD	0.06	0.64	0.58
Kreatininkinase	TZ	0.01	0.57	0.56
CHE	Harnsäure	0.17	-0.38	0.55
Temperatur	NH <sub>3</sub>	0.18	-0.36	0.54
Kreatininkinase	Cholesterin	-0.11	0.40	0.51

**Tabelle 5.4:** Vorstudien­daten: Korrelationen zwischen zwei Messvariablen mit den höchsten Korrelationsdifferenzen  $> 0.5$  zwischen Verstorbenen und Überlebenden (alle Tage, alle Patienten). Differenz bedeutet  $|\widetilde{\text{COR}}(X_s, Z_s) - \widetilde{\text{COR}}(X_d, Z_d)|$ .

2. Seien  $X_s, X_d$  und  $Z_s, Z_d$  die Mengen der Werte zweier Variablen  $X$  und  $Z$  für überlebende bzw. verstorbene Patienten. Betrachtet wurden wiederum alle Tage von allen Patienten. In der Tab. 5.4 sind die höchsten Korrelationsdifferenzen  $> 0.5$  zwischen Verstorbenen und Überlebenden verzeichnet (10 Korrelationspaare). Solch eine Betrachtung der Korrelationspaare ist sinnvoll, da vermutet wird, dass Variablen ihre Abhängigkeiten zueinander während des septischen Schocks verändern können (z.B. wie bei dem Blutdruck und der Herzfrequenz in [T<sup>+</sup>00]).

Die zur Tab. 5.4 gehörenden Anzahlen der vorhandenen Datensamples (im Fall, dass beide Variablen gleichzeitig vorhanden waren) und die Signifikanzniveaus sind in der Tab. 5.5 wiedergegeben.<sup>3</sup>

Diese Korrelationsunterschiede betreffen nur körpereigene Variablen. Wenn in den oben aufgeführten Variablen einige nicht sehr stark von Medikamenten beeinflusst werden, kann eine Korrelationsänderung zweier solcher Variablen durchaus ein Indikator für die Klasse „verstorben“ sein, da die Korrelationsänderungen sehr deutlich sind.

Um diese Korrelationsänderungen für jeden *individuellen* Patienten, sagen wir pro Stunde zu berechnen, würde man mit mind. 2 Messungen pro Stunde für die genannten Variablen nach 12h 24 Messwerte haben, mit denen man relativ sicher die Korrelationen berechnen könnte. In einem Zeitfenster von 1h könnte man dann jeweils die aktuellen Korrelationen mit zwei aktuellen Werten und

<sup>3</sup>Die Signifikanzniveaus wurden mit SPSS 9.0 (bivariate zweiseitige Pearsonsche Korrelation) berechnet.

Variable X	Variable Z	$\#(X_s, Z_s)$	$\#(X_d, Z_d)$	$s_s$	$s_d$
Lipase	Kreatinin	59	30	0.01	-
O <sub>2</sub> -Konz. Inspir.	Lipase	30	24	0.01	-
LDH	Lipase	53	25	0.01	-
KOD	Gesamtprotein	366	187	0.01	-
<b>Hämoglobin</b>	<b>CHE</b>	100	41	<b>0.01</b>	<b>0.05</b>
NH <sub>3</sub>	KOD	84	29	-	0.01
Kreatininkinase	TZ	77	26	-	0.01
CHE	Harnsäure	49	24	-	-
<b>Temperatur</b>	<b>NH<sub>3</sub></b>	424	118	<b>0.01</b>	<b>0.01</b>
Kreatininkinase	Cholesterin	39	22	-	-

**Tabelle 5.5:** Vorstudien­daten: Anzahl der Datensamples und Signifikanzniveau der Korrelationen der Tab. 5.4  $s_s := \text{Signifikanzniveau}(\widetilde{\text{COR}}(X_s, Z_s))$  und  $s_d := \text{Signifikanzniveau}(\widetilde{\text{COR}}(X_d, Z_d))$ . Gilt  $s_s \leq 0.05$  und  $s_d \leq 0.05$ , so sind die Einträge fett.

22 vorhergehenden berechnen. Die so berechneten Korrelationen könnte man mit einem neuronalen Netzwerk überwacht trainieren, um die optimale (i. Allg. nichtlineare) Klassengrenze für die verschiedenen Korrelationswerte zu finden. – Misst man die Variablen nur alle 2h, so könnte man erst nach 2 Tagen die erste Berechnung wagen und mit einem Zeitfenster von 4h arbeiten; für das medizinische Problem wäre solch ein zeitlicher Rahmen zu grob.

Leider handelt es sich bei den aufgeführten Variablen teilweise um nicht häufig gemessene Werte (Häufigkeit teilweise unter 10%), so dass dieser Vorschlag nur realisierbar wäre, falls die Messhäufigkeit seitens der Medizin deutlich verbessert werden würde. – Eine Forderung, die auf der Intensivstation im klinischen Alltag nicht erfüllbar ist. Online-monitoring könnte hier in Zukunft Abhilfe schaffen, d.h. für eine Verbesserung der Studien- und Datenanalysemöglichkeiten sorgen. – In der Tab. 5.6 (siehe auch [PHT00]) wurde nur eine Korrelationsdifferenz von mind. 0.3 gefordert; allerdings wurde zusätzlich eine minimale Messhäufigkeit von 20% gefordert. Die Ergebnisse können aufgrund der unterschiedlichen Messhäufigkeiten in der absoluten Höhe leicht verfälscht sein.

Die zur Tab. 5.6 gehörenden Anzahlen der vorhandenen Datensamples (im Fall, dass beide Variablen gleichzeitig vorhanden waren) und die Signifikanzniveaus sind in der Tab. 5.8 wiedergegeben.<sup>4</sup>

<sup>4</sup>Die Signifikanzniveaus wurden mit SPSS 9.0 (bivariate zweiseitige Pearsonsche Korrelation) berechnet.

Variable X	Variable Z	$\widetilde{\text{COR}}(X_s, Z_s)$	$\widetilde{\text{COR}}(X_d, Z_d)$	Differenz
O <sub>2</sub> -Konz. Inspir.	pH	-0.03	-0.39	0.36
Leukozyten	GGT	0.00	0.32	0.32
Eisen	GGT	0.31	0.01	0.30
Bilirubin Ges.	Harnstoff	0.26	-0.07	0.33
Harnstoff	Kreatinin	0.14	0.57	0.43
Fibrinogen	Kreatinin im Urin	0.05	-0.31	0.36
pO <sub>2</sub> Arteriell	Kalium	-0.13	0.18	0.31
TZ	Chlorid	0.24	-0.07	0.31

**Tabelle 5.6:** Vorstudiendaten: Korrelationen zwischen zwei Messvariablen mit den höchsten Korrelationsdifferenzen  $> 0.3$  zwischen  $\widetilde{\text{COR}}(X_s, Z_s)$  und  $\widetilde{\text{COR}}(X_d, Z_d)$  (alle Tage, alle Patienten). Differenz bedeutet  $|\widetilde{\text{COR}}(X_s, Z_s) - \widetilde{\text{COR}}(X_d, Z_d)|$ . Die Messhäufigkeiten der Variablen sind in der Tab. 5.7 angegeben. Es wurde mind. 20% Messhäufigkeit gefordert.

Variable	Häufigkeit [%]
Leukozyten	90.4
pH	89.0
Kreatinin	83.9
Fibrinogen	80.8
Kalium	75.3
Harnstoff	72.9
pO <sub>2</sub> Arteriell	70.4
Bilirubin Ges.	58.1
O <sub>2</sub> -Konz. Inspir.	57.3
GGT	43.6
Chlorid	37.9
TZ	36.8
Kreatinin im Urin	25.8
Eisen	20.0

**Tabelle 5.7:** Vorstudiendaten: Messhäufigkeit der Variablen aus der Tab. 5.6 (alle Patienten der Vorstudiendaten, alle Tage, kein Sampling).

Variable X	Variable Z	$\#(X_s, Z_s)$	$\#(X_d, Z_d)$	$s_s$	$s_d$
O <sub>2</sub> -Konz. Inspir.	pH	2283	1221	-	0.01
Leukozyten	GGT	1531	600	-	0.01
Eisen	GGT	665	299	0.01	-
Bilirubin Ges.	Harnstoff	1821	523	0.01	-
<b>Harnstoff</b>	<b>Kreatinin</b>	2871	682	<b>0.01</b>	<b>0.01</b>
Fibrinogen	Kreatinin im Urin	895	265	-	0.01
<b>pO<sub>2</sub> Arteriell</b>	<b>Kalium</b>	2076	568	<b>0.01</b>	<b>0.01</b>
TZ	Chlorid	676	304	0.01	-

**Tabelle 5.8:** Vorstudien­daten: Anzahl der Datensamples und Signifikanzniveau der Korrelationen der Tab. 5.6  $s_s := \text{Signifikanzniveau}(\widetilde{\text{COR}}(X_s, Z_s))$  und  $s_d := \text{Signifikanzniveau}(\widetilde{\text{COR}}(X_d, Z_d))$ . Gilt  $s_s \leq 0.05$  und  $s_d \leq 0.05$ , so sind die Einträge fett.

Aufgrund der vorgestellten Ergebnisse (Anzahl der Messungen, Signifikanzniveau) können die Paare (Harnstoff, Kreatinin) und (pO<sub>2</sub>Arteriell, Kalium) interessant für weitere (Korrelations-)Analysen sein, da z.B. gilt:

$$\widetilde{\text{COR}}(\text{Harnstoff}_s, \text{Kreatinin}_s) + \widetilde{\text{COR}}(\text{pO}_2\text{Arteriell}_s, \text{Kalium}_s) = 0.01$$

und

$$\widetilde{\text{COR}}(\text{Harnstoff}_d, \text{Kreatinin}_d) + \widetilde{\text{COR}}(\text{pO}_2\text{Arteriell}_d, \text{Kalium}_d) = 0.75.$$

Es bleibt nun Aufgabe der Medizin, durch genauere Messungen die signifikante Überprüfung an einzelnen Patienten vorzunehmen, z.B. durch 10-minütige Messungen der o.g. 4 Werte. – Ein Versuch, aufbauend auf den Korrelationsanalysen, eine Klassifikation zu erreichen, wird in [HPT<sup>+</sup>00] geschildert. Tatsächlich ist das Klassifikationsergebnis aber nicht besser als bei einer Klassifikation, die direkt auf den Variablen durchgeführt wurde.

Insgesamt muss man sagen, dass Standardanalysen die Problemlösung nicht wesentlich vorantreiben, so dass in den folgenden Abschnitten weitere Versuche durchgeführt werden. – Die Vorverarbeitungen und die Standardanalysen dienen aber sehr gut dem Prozess des Vertrautwerdens mit den Daten.

## 5.4 Zustandsanalyse

Eine Hypothese lautet, dass sich die verstorbenen Patienten während ihres Intensiv-aufenthalts in einem bislang nicht näher definierten *Zustand* „verstorben“ und die überlebenden Patienten in einem Zustand „überlebt“ befinden. Die Überprüfung dieser Hypothese kann mit überwacht lernenden neuronalen Netzen (Klassifikatoren) angegangen werden, siehe [HPT<sup>+</sup>00] und [PHT00]. Dabei beachte man den hohen Aufwand der Datenvorverarbeitung, der im Abschnitt 5.2 beschrieben wurde.

Aufbauend auf dem Abschnitt 2.3 werden hier die erzielten Ergebnisse einer Klassifikation von Patientendaten in die Zustände „kritisch“ und „nicht kritisch“ mit überwachten neuronalen Netzwerken beschrieben. Ein Klassifikationsergebnis kann man im Sinne eines Frühwarnsystems, siehe Abb. 2.2, verwenden.

Wiedergegeben werden die Ergebnisse zweier Netzwerktypen (als Vertreter *wachsender* RBF-Netze):

1. RBF-DDA (meannoise-Strategie zur fehlenden Werte-Behandlung) und
2. SGNG [HPT<sup>+</sup>00] und [PHT00] (Projektions-Strategie, s.u.).

**Hauptunterschied des SGNG im Vergleich zum RBF-DDA:** Die Zentren der radialen Basisfunktionen des SGNG sind durch einen zusätzlichen Graphen verbunden, der während des Lernvorgangs adaptiert wird. Der Graph erlaubt es, nicht nur die Parameter (Gewichte, Radien) des Gewinnerneurons zu adaptieren, sondern auch die seiner Nachbarn (=adjazente Neuronen). – Das RBF-DDA-Netz verfügt über den bereits beschriebenen Cover-Commit-Shrink-Mechanismus, der die Neuronen adaptiert. Vom Aufbau (Graph, Parameter) ist das SGNG komplexer und besitzt ein schlechteres Laufzeitverhalten (z.B. durch interne Sortiervorgänge der Knoten). Der RBF-DDA-Algorithmus garantiert im Vergleich zum SGNG die Konvergenz auf widerspruchsfreien Trainingsdaten [Ber97], erzeugt aber mehr Knoten durch die Empfindlichkeit gegenüber Ausreißern.

**Projektions-Strategie für fehlende Werte beim SGNG:** Um das SGNG aus Daten mit fehlenden Werten lernen zu lassen, kann folgendes Projektionsargument verwendet werden: Ist  $x = (x_1, \dots, x_n)^t$  ein  $n$ -dimensionaler Datenvektor, dann kann er auf den Vektor  $x$  projiziert werden, so dass kein fehlender Wert im projizierten Vektor  $x_p := (x_{i_1}, \dots, x_{i_m})^t$  vorhanden ist,  $\{i_1, \dots, i_m\} \subset \{1, \dots, n\}$ ,  $m \leq n$ ,  $x_{i_1}, \dots, x_{i_m}$  sind keine fehlenden Werte. Da das SGNG nur auf Abstandsberechnungen zwischen Vektoren basiert, ist es möglich dieses Standard-Projektionsargument auf Berechnungen in der Lern- und Anwendungsphase des SGNG zu übertragen, so dass alle Berechnungen mit den projizierten Vektoren  $x_p$  durchgeführt werden. Welcher Anteil fehlender Werte so toleriert werden kann, hängt natürlich von der Beschaffenheit des Datensatzes ab.

Maß	Mittelwert	Standardabw.	Minimum	Maximum
korr. Klass. (SGNG)	67.84	6.96	61.17	75.05
Sensitivität (SGNG)	24.94	4.85	19.38	28.30
Spezifität (SGNG)	91.61	2.53	89.74	94.49
korr. Klass. (RBF-DDA)	69.75	8.55	59.84	84.75
Sensitivität (RBF-DDA)	22.03	8.63	8.51	31.54
Spezifität (RBF-DDA)	94.19	3.84	86.82	98.15

**Tabelle 5.9:** Wichtig17Vor-Daten: Korrekte Klassifikationen auf den Testdaten mit SGNG, Sensitivität, Spezifität mit Standardabweichung, Minimum und Maximum in % von 3 (SGNG) bzw. 10 (RBF-DDA) Wiederholungen.

**Ergebnisse der Klassifikation mit SGNG:** Der verwendete Datensatz ist der „Wichtig17Vor“-Datensatz aus dem nächsten Abschnitt über Regelgenerierung (S. 161) mit den folgenden Unterschieden: Die Variablen wurden normalisiert (Mittelwert 0, Standardabweichung 1). Die Klassifikation basiert auf den 2068 Messvektoren (16-dimensional), die von den 70 septischen Schock-Patienten stammen; 348 wurden hier gelöscht aufgrund zu vieler fehlender Werte, um die o.g. Projektionsstrategie noch anwenden zu können. Mit 75% der 1720 verbleibenden Samples wurde das SGNG trainiert, und mit 25 % der Samples von *anderen* Patienten als in der Trainingsmenge wurde es getestet. 1191 der 1720 Samples (69.24%) stammen von überlebenden Patienten, 529 von verstorbenen (30.76%).

Drei Wiederholungen des gesamten Lernvorgangs mit unterschiedlichen, zufälligen Aufteilungen der Daten wurden durchgeführt. Die Ergebnisse sind in der Tab. 5.9 wiedergegeben. Um ein *allgemeingültiges* Ergebnis zu erhalten wären mehr, z.B. 10 bis 25 Wiederholungen besser; aber bei der kleinen Anzahl der Daten hätten die Ergebnisse dennoch nur einen *prototypischen*<sup>5</sup> Charakter. Man hat hier eine prototypische Fehlalarmrate von 8.39% für überlebende Patienten, die auch einen kritischen Zustand hatten, bei einer Erkennung von einem aus vier kritischen Zuständen, was ein mäßiges Ergebnis darstellt,<sup>6</sup> aber dennoch eine Hilfe für den Arzt sein kann. Erklärungen hierfür sind sicherlich die individuellen Zustände der Patienten und das retrospektiv analysierte Datenmaterial.

Ergänzung: Auf die Wiedergabe von Ergebnissen der Analysen einzelner Tage wurde in [HPT<sup>+</sup>00] verzichtet. Es wird dort in den Abschnitten 5.1.6, 5.2.6, 5.3.6 und 5.4.6 behauptet, dass die Daten keine Zeitabhängigkeit aufweisen. Es stellt sich aber heraus, dass diese Behauptung nicht zutrifft. Für mich umso mehr ein Grund im Kapitel 7 die Zeitabhängig-

<sup>5</sup>D.h. die Ergebnisse erhalten eine signifikante Bedeutung erst bei einer Überprüfung mit weiteren Daten. Leider standen keine weiteren Ergebnisse für das SGNG zur Verfügung.

<sup>6</sup>Insbesondere liegt die Leistung nur im Bereich des trivialen Klassifikators, der allerdings keine Zustände kritischer Patienten entdecken würde.

keit auf den MEDAN-Daten zu berücksichtigen. Die Berücksichtigung der Zeitabhängigkeit wird zu einer Verbesserung der Analyseergebnisse führen.

**Ergebnisse der Klassifikation mit RBF-DDA:** Der verwendete Datensatz ist der o.g. „Wichtig17Vor“-Datensatz. Es wurde wiederum eine Aufteilung der Daten in 75% Trainingsdaten und 25% Testdaten vorgenommen. Keine Daten eines Patienten aus der Testmenge sind in der Trainingsmenge enthalten. Allerdings stammen die Ergebnisse aus 10 Versuchswiederholungen, siehe die Tab. 5.9.

**Vergleich der Ergebnisse zwischen SGNG und RBF-DDA:** Im Mittel sind die Klassifikationsergebnisse fast identisch. Die Standardabweichungen sind beim RBF-DDA größer mit kleineren Minimalwerten und größeren Maximalwerten. Da beim RBF-DDA 10 Versuchswiederholungen und beim SGNG nur 3 Versuchswiederholungen durchgeführt wurden, ist es wahrscheinlich, dass bei den 10 Wiederholungen ein kleineres Minimum und ein größeres Maximum aufgetreten ist, so dass die Unterschiede nicht eindeutig zu beurteilen sind. Bei den Standardabweichungen, die bei diesem Datensatz aufgetreten sind, müsste man in der Anwendung bei beiden Netzen sehr sorgfältig ein geeignetes auswählen, um die Extrema zu vermeiden.

**Ergebnisse einer Clusteranalyse:** Die Ergebnisse einer ergänzend durchgeführten (hierarchischen) *Clusteranalyse* kann man in [Pau01] nachlesen; allerdings ergeben sich (wie bei Versuchen mit der U-Matrix-Methode, die z.B. mit den concensus-conference-Variablen systolischer Blutdruck, Leukozyten, Herzfrequenz und Beatmungsfrequenz<sup>7</sup> durchgeführt wurde; die hier allerdings aufgrund des schlechten Ergebnisses nicht dokumentiert sind) keine eindeutigen, verwertbaren Ergebnisse. Das kann an der geringen Datenanzahl und der nicht sehr genau (bez. der Zeitstempel) durchgeführten Messungen liegen. Man darf aber nicht ausschließen, dass die Daten des septischen Schocks auch bei günstigerer Datenlage nicht ausschließlich durch eine (unüberwachte) Clusteranalyse einiger Trenddaten in die Zustände „kritisch“ und „nicht kritisch“ eingeteilt werden können.

Insgesamt liefern die so durchgeführten Zustandsanalysen noch kein eindeutiges Bild. Insbesondere ist nicht klar, wo besser zu klassifizierende Regionen liegen und wo schlechter zu klassifizierende Regionen liegen. Die Regelgenerierung der nächsten Abschnitte bietet einige Erklärungen. Eine Zusammenfassung der Zustandsanalyse und der Regelgenerierung kann man auch in [BHP02] nachlesen. Verbessern wird sich die Analysesituation im 7. Kapitel, wo konsequent die Zeitabhängigkeit der Analyse mit untersucht wird.

---

<sup>7</sup>Die Atemfrequenz stand für eine Analyse nicht zur Verfügung.

## 5.5 Regelgenerierung

Aufbauend auf den Abschnitten 2.2.3, 2.5 und dem Kapitel 3 werden nun die Verfahren zur Regelgenerierung angewendet. Im nächsten Unterabschnitt findet der Fuzzy-RecBF-DDA-Regelgenerator aus dem Abschnitt 3.1 Verwendung. – Im darauffolgenden Unterabschnitt wird ein interessantes Experiment unternommen. Die im Projekt mitarbeitenden Mediziner wurden gebeten, Regeln aus ihrem Erfahrungsschatz heraus aufzustellen, was sie als sehr schwierig erachteten. Diese wurden anschließend anhand der gesammelten Daten auf ihre Qualität hin überprüft und mit den automatisch generierten Regeln verglichen.

Im Folgenden werden die wesentlichen Datensätze  $\mathcal{DS}$  beschrieben, mit denen eine Regelgenerierung auf den Vorstudien Daten durchgeführt wurde:

**Datensatz „Häufig12Vor“:** Der Datensatz besteht aus den 12 am häufigsten gemessenen Variablen der Vorstudie (septische Schock-Patienten, alle Tage, 24h-Sampling) gemäß der Tab. 5.2, d.h.<sup>8</sup> Kreatinin [mg/dl], Calcium [mmol/l], pCO<sub>2</sub>Arteriell [mmHg], pH [-], Haematokrit [%], Natrium [mmol/l], Leukozyten [1000/ $\mu$ l], Haemoglobin [g/dl], ZVD [cmH<sub>2</sub>O], Temperatur [°C], Herzfrequenz [/min], BlutdruckSystolisch [mmHg]. Von den insgesamt 2068 Samples waren 490 nicht mit allen 12 Variablen besetzt, 399 nicht mit mind. 11 Variablen, 370 nicht mit mind. 10 Variablen, ..., 336 nicht mit mind. 1 Variable besetzt. Um möglichst viele Samples zu haben, aber die Qualität der Daten nicht zu stark zu mindern, wurde hier gefordert, dass mind. 10 der 12 Variablen pro Sample vorhanden waren, d.h. es stehen 1698 Samples zur Verfügung.

**Datensatz „Wichtig17Vor“:** Der Datensatz besteht aus 17 von Medizinern als „wichtig“ eingestuften Variablen der Vorstudie (septische Schock-Patienten, alle Tage, 24h-Sampling), namentlich Herzfrequenz [/min], BlutdruckSystolisch [mmHg], pO<sub>2</sub>Arteriell [mmHg], pCO<sub>2</sub>Arteriell [mmHg], Beatmungsfrequenz [/min], O<sub>2</sub>KonzentrationInspiratorisch [%], pH [-], Leukozyten [1000/ $\mu$ l], Thrombozyten [1000/ $\mu$ l], TPZ [%], AT3 [%], Laktat [mg/dl], Kreatinin [mg/dl], Urinmenge [ml/24h], DopaminHochdosierte [μg/kg/min], DopaminNiedrigdosierte [μg/kg/min], Dobutrex [μg/kg/min]. Von den insgesamt 2068 Samples war kein Sample mit allen 17 Variablen besetzt, 2053 nicht mit mind. 16 Variablen, 1823 nicht mit mind. 15 Variablen, ..., 430 nicht mit mind. 11 Variablen, 391 nicht mit mind. 10 Variablen, 370 nicht mit mind. 9 Variablen, ..., 329 nicht mit mind. 1 Variable besetzt. Um möglichst viele Samples zu haben, aber die Qualität der Daten nicht zu stark zu mindern, wurde hier gefordert, dass mind. 10 der 17 Variablen pro Sample vorhanden waren, d.h. es stehen 1677 Samples zur Verfügung. Die Werte der Variablen DopaminHochdosierte und DopaminNiedrigdosierte wurden addiert und mit „Dopamin Hoch+Niedrig“ bezeichnet, so dass mit 16 Variablen trainiert wurde. Die Werte von Dopamin Hoch+Niedrig

---

<sup>8</sup>Die Einheiten stehen in eckigen Klammern.

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	72.53	4.06	66.76	76.50
Training nicht klassifiziert	0.03	0.06	0.00	0.13
Test korrekt	67.71	5.51	63.31	77.07
Test nicht klassifiziert	0.07	0.10	0.00	0.22
Sensitivität	57.95	10.17	42.64	65.46
Spezifität	70.21	5.47	64.12	77.78

**Tabelle 5.10:** Häufig12Vor-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Sensitivität und Spezifität auf den Testdaten (ohne die nicht klassifizierten Testdaten). Angaben in Prozent.

und Dopamin wurden einer speziellen fehlende Werte-Behandlung unterzogen: War kein Wert vorhanden, so wurde er auf Null gesetzt, da für diese Medikamente davon ausgegangen werden konnte, dass kein Medikament gegeben wurde.

Die zufällig gewählten Test- und Trainingsdaten kommen in den Experimenten jeweils von unterschiedlichen Patienten. **Klasse ü** bezeichnet immer die Klasse der Daten von *überlebenden* Patienten und **Klasse v** die Klasse der Daten der *verstorbenen*.

### 5.5.1 Regelgenerierung mit Fuzzy-RecBF-DDA-Netzen

Als Erstes werden die Versuchseinstellungen wiedergegeben; anschließend werden die Ergebnisse dargestellt und diskutiert. Für die Versuche wird, wenn nicht anders angegeben, das von mir modifizierte Fuzzy-RecBF-DDA-Verfahren verwendet, das kurz mit „Fuzzy-RecBF-DDA“ bezeichnet wird.

**Versuchseinstellungen für Häufig12Vor:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 65$ , Ergebnisse siehe Tabellen 5.10 und 5.11 (Klassifikation) und Tabellen 5.12, C.17 und C.18 (Regelgenerierung). Die verwendeten Testdaten kamen in den 5 Testdurchläufen im Mittel zu 67.63% aus Klasse ü und zu 32.37% aus Klasse v. Die Epochenanzahl betrug im Mittel 6.2 mit einer Standardabweichung von 1.79 (Minimum 3, Maximum 7). Die Testdaten stammen von 35 Patienten, die disjunkt zu den 35 Patienten der Trainingsmenge sind.

Um das Klassifikationsergebnis in Hinblick auf die Veränderung der Klassifikationsschwelle besser einschätzen zu können, werden (wie auch schon im Kapitel 3) in der Abb. B.3 alle Klassifikationsergebnisse der jeweils 5 Versuche der Häufig12Vor-Daten und der Wichtig17Vor-Daten aufgetragen. Es wurden jeweils 10 Punkte eingetragen, also wurden insgesamt 2 mal 50 Klassifikationen durchgeführt. Berechnet man den mittleren Flächeninhalt unter den einzelnen ROC-Kurven, so erhält

	Klasse ü	Klasse v
Klasse ü (Training)	65.02	21.41
Klasse v (Training)	6.04	7.50
Klasse ü (Test)	61.91	26.54
Klasse v (Test)	5.68	5.80

**Tabelle 5.11:** Häufig12Vor-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

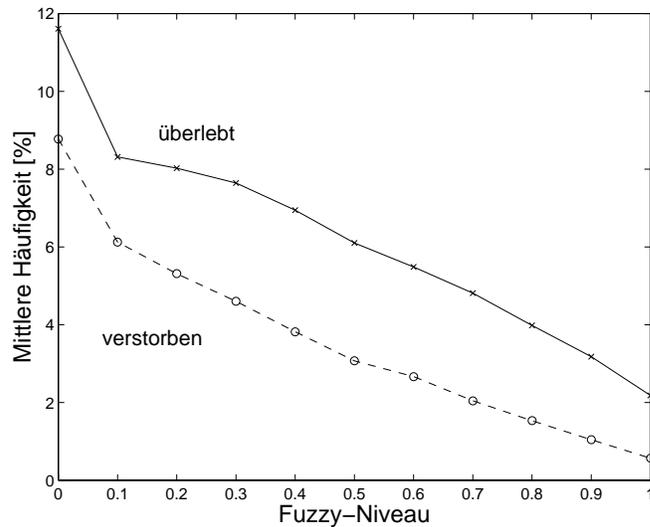
	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	31.8	2.2	30	35
Klasse v	30.6	9.6	23	42

**Tabelle 5.12:** Häufig12Vor-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

man für die Wichtig17Vor-Daten den Wert 0.577 (Min.: 0.506, Max.: 0.664) und für die Häufig12Vor-Daten den Wert 0.608 (Min.: 0.566, Max.: 0.723). Die Schwankungsbreite deutet auf eine stärkere Abhängigkeit der Daten von der Einteilung in Trainings- und Testdaten hin, d.h. für signifikantere Ergebnisse werden sehr viel mehr Daten nötig sein. Die Flächeninhalte sind naturgemäß nicht so hoch, da hier die Samples der Zeitreihe klassifiziert werden und nicht die Patienten wie im Abschnitt 6.5, wo bessere Ergebnisse erzielt wurden.

**Häufig12Vor-Daten (Fuzzy-Regeln zu den Niveaus 0.0 bis 1.0):** In der Abb. 5.2 (Klasse ü = überlebt, Klasse v = verstorben) sieht man die abnehmende mittlere Häufigkeit der Regeln, wenn das Fuzzy-Niveau erhöht wird; das ist ein normales Kurvenverhalten. Für diesen hochdimensionalen Datensatz sind die mittleren Häufigkeits-Werte für die Support-Regeln noch einigermaßen hoch und damit durchaus medizinisch verwertbar. Sie nehmen allerdings bis zum Core-Niveau 1.0 deutlich ab. In dieser Größenordnung sind die Regeln nicht mehr medizinisch verwertbar.

Leider ergibt sich bei der mittleren Konfidenz ein sehr unbefriedigendes Bild (Abb. 5.3). Erwartet man für einen Datensatz mit gut trennbaren Klassen (z.B. IRIS-Daten) eine Zunahme der mittleren Konfidenz bis zu einem gewissen Niveau nahe bei 1, so sieht man hier, dass ab dem Niveau 0.1 keine wesentliche Zunahme zu beobachten ist, ab dem Niveau 0.1 sogar bei der Klasse v eine Abnahme. (Man erinnere sich, dass die Konfidenz für einzelne Regeln auf 0 gesetzt wird, falls keine Testdaten in den Regelbereich fallen.) Insgesamt liegen die maximalen Werte etwa im Bereich der prozentualen Aufteilung der Testdaten: Klasse ü: mittlere Konfidenz zwischen 59.21% (Niveau 0.9) und 65.83% (Niveau 0.0), Testdatenaufteilung:



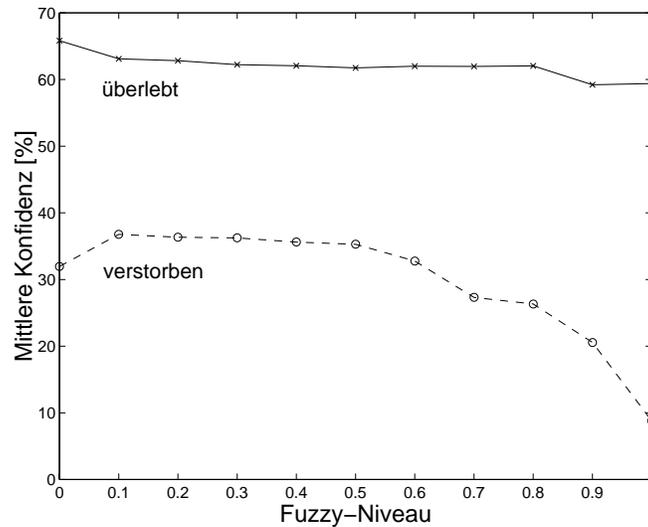
**Abbildung 5.2:** Häufig12Vor-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

67.63%; Klasse v: mittlere Konfidenz zwischen 8.82% (Niveau 1.0) und 36.77% (Niveau 0.1), Testdatenaufteilung: 32.37%.

**Versuch mit Fuzzy-WvRecBF-DDA (Häufig12Vor-Daten):** Hier wird noch ein Versuch mit den Häufig12Vor-Daten wiedergegeben, der mit dem Fuzzy-WvRecBF-DDA durchgeführt wurde. Hier wurde auch der nearby-Parameter (nearby = 0.54,  $a = 0$ ) verwendet.

Dazu sei die vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 65$ , Klassifikationsergebnis siehe Tab. 5.13, Anzahl der erzeugten Regeln siehe Tab. 5.14. Die verwendeten Testdaten kamen in den 7 Testdurchläufen im Mittel zu 70.57% aus Klasse  $\bar{u}$  und zu 29.43% aus Klasse  $v$ . Die Epochenzahl betrug im Mittel 5.6 mit einer Standardabweichung von 1.81 (Minimum 3, Maximum 7). Die Testdaten stammen von 35 Patienten, die disjunkt zu den 35 Patienten der Trainingsmenge sind. Weitere Parametereinstellungen:  $\text{min}_{dp} = 3$  und  $s_{\text{weak}} = 0.3$ .

Das Klassifikationsergebnis ist ähnlich zu dem bei der Anwendung des Fuzzy-RecBF-DDA erzielten Ergebnisses. Aufgrund des nearby-Parameters wurden mehr Regeln erzeugt. Auf die Angabe der Konfidenz- und Häufigkeits-Tabellen verzichte ich an dieser Stelle, da sie ebenfalls ähnliche Werte liefern. Die Ergebnisse profitieren nicht so stark von den Verbesserungen des Fuzzy-WvRecBF-DDA-Algorithmus, weil die Daten einen zu hohen Zufallscharakter besitzen und damit auch nicht unbedingt repräsentativ für das Problem der Ausreißer sind. Die Fuzzy-Niveau-Regeln zum Niveau 0.4 hatten die beste Güte im Versuch mit dem Fuzzy-WvRecBF-DDA.



**Abbildung 5.3:** Häufig12Vor-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	74.62	3.82	68.84	78.66
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	69.00	4.37	61.20	72.66
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 5.13:** Häufig12Vor-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten (Fuzzy-WvRecBF-DDA). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	43.4	4.3	38	50
Klasse v	48.4	3.9	44	55

**Tabelle 5.14:** Häufig12Vor-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum) (Fuzzy-WvRecBF-DDA).

Nr.	Variable	überlebt	verstorben	Summe
1	Herzfrequenz	0.304	0.197	0.501
2	BlutdruckSystolisch	0.401	0.134	0.535
3	ZVD	<b>0.664</b>	0.424	<b>1.088</b>
4	pCO <sub>2</sub> Arteriell	0.298	0.095	0.393
5	pH	0.314	<b>0.594</b>	0.908
6	Temperatur	0.375	0.400	0.775
7	Leukozyten	0.316	0.234	0.550
8	Hämoglobin	0.010	0.003	0.013
9	Hämatokrit	0.188	0.046	0.234
10	Kreatinin	0.127	0.166	0.293
11	Natrium	0.515	0.453	0.968
12	Calcium	0.155	0.111	0.266

**Tabelle 5.15:** Häufig12Vor-Daten: Wichtigkeit der Variablen bez. den Klassen „überlebt“ und „verstorben“, gemittelt über alle Versuchswiederholungen. Der höchste Wert pro Spalte ist fett markiert.

**Fazit:** Bei den Versuchen mit den Häufig12Vor-Daten ist die mittlere Konfidenz nicht gut im Vergleich zur Testdatenaufteilung. – Nach der Untersuchung der Wichtigkeit und Frühzeitigkeit wird im Folgenden ein weiterer Datensatz analysiert, der aus den von Medizinern als „wichtig“ deklarierten Variablen besteht und bei dem die Ergebnisse etwas besser, aber nicht entscheidend besser sind. Da die Fuzzy-WvRecBF-DDA-Variante keine überzeugenderen Resultate geliefert hat, wird sie auf dem nächsten Datensatz nicht mehr verwendet.

**Wichtigkeit und Frühzeitigkeit:** Kommen wir nun zur Wichtigkeit der Variablen des Datensatzes Häufig12Vor, die im Abschnitt 3.5 definiert wurde. In der Tab. 5.15 sieht man, dass die Wichtigkeit für die Klasse „überlebt“ bei den Variablen 3 und 11 hoch ist. Die Variablen 3, 5, 6 und 11 haben eine hohe Wichtigkeit bez. der Klasse „verstorben“. In der Summe haben die Variablen 3, 5 und 11 eine hohe Wichtigkeit. Geringe bzw. keine Aussagekraft hat die Variable 8. Die anderen Variablen haben Werte dazwischen; außer Hämoglobin ist keine Variable unwichtig.

Die Frühzeitigkeit der erzeugten (Support-)Regeln ergibt hier ein unbefriedigendes Bild: Klasse „überlebt“ Mittelwert 27.7, Maximum 29.9, Minimum 23.6, Standardabweichung 1.6 [Tage vor dem letzten Tag des Intensivaufenthalts]; Klasse „verstorben“ Mittelwert 26.7, Maximum 29.7, Minimum 23.2, Standardabweichung 1.6 [Tage vor dem letzten Tag des Intensivaufenthalts].<sup>9</sup> Die geringe Standardabweichung und die geringen Unterschiede zwischen Überlebenden und Verstorbenen besagen, dass im

<sup>9</sup>Die Liegedauer der Patienten beträgt im Mittel 29.5 Tage.

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	69.17	9.98	59.73	83.94
Training nicht klassifiziert	0.09	0.12	0.00	0.24
Test korrekt	68.42	8.79	52.92	74.74
Test nicht klassifiziert	0.10	0.22	0.00	0.48
Sensitivität	36.45	11.66	16.85	47.06
Spezifität	74.28	12.11	53.06	81.73

**Tabelle 5.16:** Wichtig17Vor-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Sensitivität und Spezifität auf den Testdaten (ohne die nicht klassifizierten Testdaten). Angaben in Prozent.

Mittel die Regeln zu keinem bestimmten Zeitpunkt auftreten, z.B. Klasse „überlebt“ eher früher, Klasse „verstorben“ eher später. Das liegt wohl daran, dass der Zustand eines Patienten häufig wechselt; bei einigen Patienten treten die Regeln früher auf, bei anderen später. Man erhält diesbez. keine brauchbare Aussage. Unangenehm ist auch die unterschiedliche Liegedauer der Patienten (1 bis 122 Tage), da die Frühzeitigkeit idealerweise von möglichst gleichlangen Zeitreihen ausgeht. Auch ist der Wert durch die fehlenden Werte innerhalb eines Samples verfälscht, da sie bei der Berechnung der Frühzeitigkeit immer als „zutreffend“ gewertet wurden. Insgesamt ist es hier sinnvoll, keine Schlüsse aus der Frühzeitigkeit zu ziehen.

Neben den häufigen Variablen erscheint es sinnvoll, die *von Medizinern* als wichtig eingestuften Variablen zu analysieren.

**Versuchseinstellungen für Wichtig17Vor:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 65$ , Ergebnisse siehe Tabellen 5.16 und 5.17 (Klassifikation) und Tabellen 5.18, C.19 und C.20 (Regelgenerierung). Die verwendeten Testdaten kamen in den 5 Testdurchläufen im Mittel zu 72.10 % aus Klasse  $\ddot{u}$  und zu 27.91 % aus Klasse  $v$ . Die Epochenzahl betrug im Mittel 4.0 mit einer Standardabweichung von 1.73 (Minimum 3, Maximum 7). Die Testdaten stammen von 35 Patienten, die disjunkt zu den 35 Patienten der Trainingsmenge sind.

Die ROC-Kurven für den Wichtig17Vor-Datensatz findet man (zusammen mit denen für den Häufig12Vor-Datensatz) in der Abb. B.3.

**Wichtig17Vor-Daten (Fuzzy-Regeln zu den Niveaus 0.0 bis 1.0):** In der Abb. 5.4 (Klasse  $\ddot{u}$  = überlebt, Klasse  $v$  = verstorben) sieht man die abnehmende mittlere Häufigkeit der Regeln, wenn das Fuzzy-Niveau erhöht wird, ein normales Verhalten. Für diesen hochdimensionalen Datensatz sind die mittleren Häufigkeitswerte für die Support-Regeln noch recht hoch und damit durchaus medizinisch

	Klasse ü	Klasse v
Klasse ü (Training)	59.75	25.35
Klasse v (Training)	5.39	9.42
Klasse ü (Test)	63.35	8.67
Klasse v (Test)	22.82	5.06

**Tabelle 5.17:** Wichtig17Vor-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

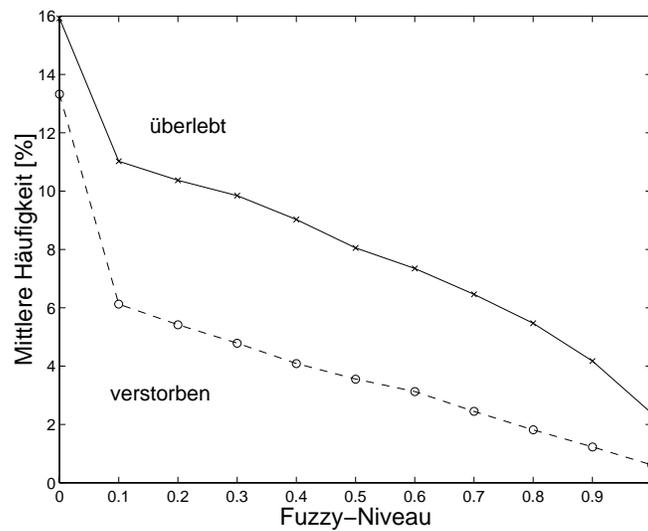
	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	22.8	5.5	18	31
Klasse v	17.8	3.9	12	22

**Tabelle 5.18:** Wichtig17Vor-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

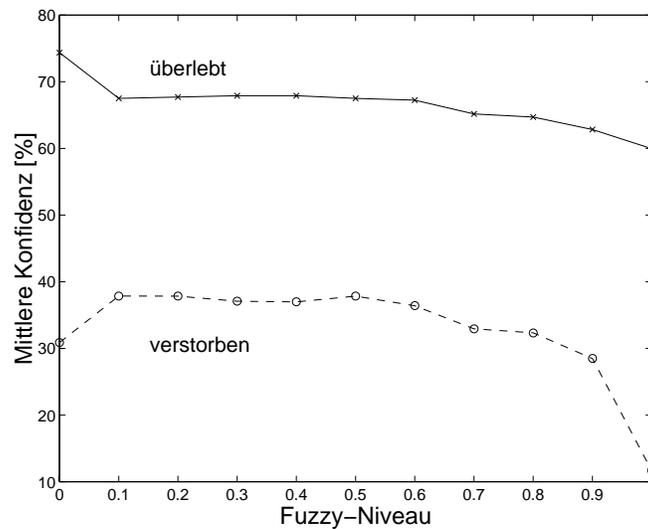
verwertbar. Sie nehmen allerdings bis zum Core-Niveau 1.0 deutlich ab. In dieser Größenordnung sind die Regeln nicht mehr medizinisch verwertbar. Das ist ein ähnliches Ergebnis wie für den Häufig12Vor-Datensatz.

Leider ergibt sich bei der mittleren Konfidenz ein unbefriedigendes Bild (Abb. 5.5). Bei einem Datensatz mit gut trennbaren Klassen erwartet man eine Zunahme der mittleren Konfidenz bis zu einem gewissen Niveau nahe bei 1; hier sieht man, dass ab dem Niveau 0.1 keine bzw. eine sehr geringe Zunahme zu beobachten ist, ab dem Niveau 0.6 eine Abnahme, die insbesondere für die Klasse v (verstorben) stark ausfällt. (Man erinnere sich, dass die Konfidenz für einzelne Regeln auf 0 gesetzt wird, falls keine Testdaten in den Regelbereich fallen.)

Insgesamt liegen die maximalen Werte etwa im Bereich der prozentualen Aufteilung der Testdaten: Klasse ü: mittlere Konfidenz zwischen 59.96% (Niveau 1.0) und 74.37% (Niveau 0.0), Testdatenaufteilung: 72.10%; Klasse v: mittlere Konfidenz zwischen 11.70% (Niveau 1.0) und 37.85% (Niveau 0.1), Testdatenaufteilung: 27.91%. Das ist ebenfalls ein ähnliches Ergebnis wie für den Häufig12Vor-Datensatz.



**Abbildung 5.4:** Wichtig17Vor-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 5.5:** Wichtig17Vor-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

**Fazit:**

Hier kann man nur insofern von einem medizinisch verwertbaren Ergebnis sprechen, insofern man akzeptiert, dass die Daten *global nicht* durch Regeln zu erklären sind. *Lokal* ergeben sich einige interessante Regeln, von denen exemplarisch zwei wiedergegeben werden (zum Support-Niveau 0.0, aus dem zweiten von fünf Durchläufen), d.h. es handelt sich um Regeln, die eine bessere Performanz als der Durchschnitt haben.

Das im Vergleich zu den „häufigen“ Variablen etwas bessere Ergebnis kann auf die höhere Dimension (bessere Trennbarkeit) oder auf die zusätzliche Verwendung zweier zusätzlicher Medikamente zurückgeführt werden.

Hier nun die zwei Beispiele für Regeln (6. und 43. Regel aus dem 2. Versuch):

**6)** support class **survived** with test frequency **0.092** and test confidence **0.912**  
 from **11** different patients if  
**var** *Herzfrequenz*  $\geq 105.00$  and  
**var** *BlutdruckSystolisch*  $\geq 130.00$  and  
**var** *O<sub>2</sub>KonzentrationInspiratorisch*  $\leq 60.00$  and  
**var** *Beatmungsfrequenz*  $\geq 19.00$  and  
**var** *Leukozyten*  $\leq 16.70$  and  
**var** *Dobutrex*  $\leq 1.50$

**43)** support class **deceased** with test frequency **0.076** and test confidence **0.697**  
 from **13** different patients if  
**var** *BlutdruckSystolisch*  $\geq 120.00$  and  
**var** *Leukozyten*  $\geq 24.10$  and  
**var** *Dobutrex*  $\leq 6.00$

**Wichtigkeit und Frühzeitigkeit:** Kommen wir nun zur (über alle Versuchswiederholungen gemittelten) Wichtigkeit der Variablen des Datensatzes Wichtig17Vor, die im Abschnitt 3.5 definiert wurde. In der Tab. 5.19 sieht man, dass die Wichtigkeit für die Klasse „überlebt“ für die Variablen 5, 9 und 15 eher hoch ist, für Variable 15 sogar deutlich höher. Für die Klasse „verstorben“ haben die Variablen 2, 5 und 15 höhere Werte, die sich aber nicht deutlich von den anderen abheben. In der Summe hat die Variable 15 den höchsten Wert. Geringe Aussagekraft haben die Variablen 11, 12 und 14. Die Werte für die anderen Variablen liegen dazwischen. Insgesamt ergibt sich kein eindeutiges Bild, die zu einer *deutlichen* Verringerung der Variablenanzahl führen kann, wie das z.B. beim Vote-Datensatz (Tab. 4.1) der Fall gewesen ist; allerdings werden im Kap. 7 auf den Klinikdatensätzen noch detailliertere Untersuchungen vorgenommen.

Nr.	Variable	überlebt	verstorben	Summe
1	Herzfrequenz	0.330	0.127	0.457
2	BlutdruckSystolisch	0.292	<b>0.315</b>	0.607
3	pO <sub>2</sub> Arteriell	0.134	0.082	0.216
4	pCO <sub>2</sub> Arteriell	0.118	0.032	0.150
5	O <sub>2</sub> Konz.Inspir.	0.400	0.238	0.638
6	Beatmungsfrequenz	0.198	0.165	0.363
7	pH	0.110	0.042	0.152
8	Leukozyten	0.342	0.198	0.540
9	Thrombozyten	0.486	0.120	0.606
10	TPZ	0.201	0.160	0.361
11	AT3	0.033	0.068	0.101
12	Laktat	0.010	0.019	0.029
13	Kreatinin	0.028	0.129	0.157
14	Urinmenge	0.073	0.018	0.091
15	Dopamin Hoch+Niedrig	<b>0.790</b>	0.281	<b>1.071</b>
16	Dobutrex	0.311	0.134	0.445

**Tabelle 5.19:** Wichtig17Vor-Daten: Wichtigkeit der Variablen bez. den Klassen „überlebt“ und „verstorben“. Der höchste Wert pro Spalte ist fett markiert.

Die Frühzeitigkeit der erzeugten Regeln ergibt auch bei diesem Datensatz ein unbefriedigendes Bild: Klasse „überlebt“ Mittelwert 26.2, Maximum 30.5, Minimum 20.9, Standardabweichung 2.1 [Tage vor dem letzten Tag des Intensivaufenthalts]; Klasse „verstorben“ Mittelwert 25.9, Maximum 29.1, Minimum 21.7, Standardabweichung 2.1 [Tage vor dem letzten Tag des Intensivaufenthalts]. Die Erklärung für die Ergebnisse ist dieselbe wie beim Datensatz Häufig12Vor auf der S. 166.

#### **Fazit:**

Die Ergebnisse zeigen a) dass aufgrund der Datenqualität die Ergebnisse nicht hervorragend sind und b) dass es sinnvoll ist, bei den qualitativ besseren Klinikdaten die Zeitabhängigkeit zu berücksichtigen (siehe Kap. 7).

Die Analyse der Vorstudien Daten hat Einblicke in die komplizierte Struktur der septischer Schock-Daten erlaubt. Wie kompliziert die Einschätzung der Daten ist, belegt das Experiment im folgenden Abschnitt.

### **5.5.2 Manuelle und automatische Regelgenerierung**

Ein interessanter Ansatz bez. der Regelgenerierung ist es, einen medizinischen Experten Regeln vorgeben zu lassen, sie auszuwerten und mit den automatisch erzeugten zu vergleichen.

Ein Regelvorschlag eines erfahrenen Mediziners lautet:

**Med 1)**

```
class deceased if
var  $pH \leq 7.2$  and
var  $pO_2Arteriell \leq 60$  and
var  $O_2KonzentrationInspiratorisch \geq 80$  and
var  $BaseExcess \leq 5$ 
```

Ausgehend von 2068 Messungen (septische Schock-Patienten der Vorstudie, alle Tage, 24h-Sampling) gab es 943 Messungen, bei denen die Werte pH,  $pO_2Arteriell$ ,  $O_2KonzentrationInspiratorisch$  und  $BaseExcess$  gleichzeitig gemessen wurden. Unter diesen Messungen gibt es *keine*, die diesen Regelvorschlag erfüllt, d.h. kein Patient hat je diese Daten vorweisen können. (Ergänzung: Das gilt auch, wenn man statt den 24h-Sampling-Daten die Rohdaten betrachtet.) Unter den 2068 Messungen gab es zwar 22, bei denen  $pH \leq 7.2$  gilt, aber in diesen Fällen wurde nie  $BaseExcess$  gemessen (fehlende Werte).

Um dennoch eine Auswertung machen zu können, wurde der Konflikt aufgelöst, indem nur drei der o.g. Variablen gleichzeitig betrachtet wurden:

**Med 1.1)**

```
class deceased if
var  $pH \leq 7.2$  and
var  $pO_2Arteriell \leq 60$  and
var  $O_2KonzentrationInspiratorisch \geq 80$ 
```

**Med 1.2)**

```
class deceased if
var  $pO_2Arteriell \leq 60$  and
var  $O_2KonzentrationInspiratorisch \geq 80$  and
var  $BaseExcess \leq 5$ 
```

Zur Regel 1.1 existieren nur 2 Messwerte (von 1242, bei denen diese drei Werte gleichzeitig erfasst sind), das sind 0.2% der Daten. Diese beiden Messwerte stammen beide von Verstorbenen und zwar von zwei verschiedenen Personen, d.h. die Test-Häufigkeit ist 0.002 und die Test-Konfidenz ist 1.000.

Zur Regel 1.2 existieren nur 3 Messwerte (von 944, bei denen diese drei Werte gleichzeitig erfasst sind), das entspricht 0.3% der Daten. Von diesen drei Messwerten stammt einer von einem Verstorbenen und zwei stammen von (verschiedenen) Überlebenden, d.h. die Test-Häufigkeit ist 0.003 und die Test-Konfidenz ist 0.333.

**Fazit:** Die drei analysierten Regeln sind weit davon entfernt, die Verstorbenen oder einen großen Teil der Verstorbenen zu charakterisieren. Der Grund liegt einfach darin, dass diese Regeln auf keine bzw. auf eine nur sehr geringe Datenmenge Anwendung finden. Regel 1.2 trifft sogar mehrheitlich auf Daten von Überlebenden zu. Man kann wohl in dieser Situation nicht einfach sagen: Wenn alle diese Variablen gleichzeitig „schlechte“ Werte annehmen, stirbt der Patient. Wahrscheinlicher ist, dass immer nur ein Teil der Variablen gleichzeitig „schlecht“ ist. Der Experte liegt also mit dieser Regel falsch. Allerdings ergaben diese 4 Variablen auch mit den automatischen Regelgewinnungsalgorithmen kein brauchbares Ergebnis auf den Vorstudientaten für Verstorbene, d.h. mit diesen 4 Variablen ist eine Regelgenerierung bzw. Klassifikation nicht sinnvoll.

Da die manuelle Regelgenerierung schwierig ist, ist es umso wichtiger, die maschinengetriebene Analyse fortzusetzen.



# Kapitel 6

## Ausgewählte symbolische septischer Schock-Klinikdaten

Im vorherigen Kapitel wurden die metrischen Daten der Vorstudie mit einem metrischen Regelgenerator untersucht und quantitative Ergebnisse vorgestellt. Da der Zugriff auf die symbolischen Vorstudien­daten technisch nicht möglich war, aber im Anschluss auch eine Analyse der symbolischen Klinikdaten wünschenswert ist, wurden die symbolischen Daten einer ausgewählten Klinik für eine Analyse ausgesucht („ASK-Daten“, vgl. den Abschnitt A.4). Wie auch die Analysen im letzten Kapitel sollen auch diese Analysen der Evaluierung der Verfahren an den medizinischen Daten der septischen Schock-Patienten dienen und grundsätzliche Einblicke in das Datenmaterial liefern.

Nach einer kurzen Beschreibung der verwendeten Daten im Abschnitt 6.1, werden die Ergebnisse der Regelgenerierung wiedergegeben und diskutiert (Abschnitt 6.2). Die Analysen beinhalten neben der Klassifikation und Regelgenerierung wieder die Untersuchung der *Wichtigkeit*, der *Frühzeitigkeit* und der *Ähnlichkeit* (Abschnitte 6.2 bis 6.4). Im Abschnitt 6.5 wird die Verwendbarkeit der *Generalisierungsindices* untersucht. Zusätzlich werden Versuche, die auf einem *Frequent Patterns-Baum* bzw. auf *Assoziationsregeln* beruhen, durchgeführt (Abschnitt 6.6). Ausgewählte Inhalte dieses Kapitels findet der Leser auch in [PB02].

### 6.1 Beschreibung der verwendeten Daten

Alle Variablen wurden zeitunabhängig binär kodiert, siehe den Abschnitt A.4. Es handelt sich um Daten von 362 septische Schock-Patienten der Abdominalchirurgie, von denen 54 (14.9%) verstorben sind. Im Vergleich zu den Vorstudien- und Klinikdaten ist die prozentuale Anzahl der Verstorbenen niedriger, was auf eine etwas andere Handhabung des septischer Schock-Kriteriums zurückzuführen ist (Nachweis der Infektion nicht gefordert) oder aber auch zusätzlich auf probabilistische Schwankungen.

Von den 48 binären Variablen (gleich 96 Items) waren für jeden Patienten Werte von mindestens 12 (25.0%) und maximal allen 48 Variablen vorhanden. Im Mittel waren 41 (85.4%) Variablen pro Patient vorhanden. Da Generalisierungsregeln, die Variablen enthalten, bei denen häufig Werte fehlen, selbst nicht häufig sein können, werden aufgrund der fehlenden Werte keine fälschlicherweise als interessant erachteten Regeln erzeugt. Natürlich besteht die Möglichkeit, dass einige Regeln deshalb nicht erzeugt werden.

## 6.2 Anwendung des Algorithmus zur Erzeugung von Generalisierungsregeln

Ausgehend von den 362 Patienten, deren Variablenwerte als Itemsets kodiert wurden, wurden drei Versuchswiederholungen durchgeführt. Die Heuristiken 1., 2. und 4. aus dem Abschnitt 4.6 wurden angewendet. Für das erste Level wurden 50% der Daten zum Training verwendet, die anderen 50% zum Testen, u.a zur Bestimmung der Klassifikationsleistung. Es lagen jeweils folgende Einstellungen vor:  $\max_{level} = 5$ ,  $\max_{new} = 3$ ,  $\min_{freq} = 0.34$  sowohl für die Trainings- als auch die Test-Itemsets für die Klasse „überlebt“ (Klasse  $\ddot{u}$ ) und  $\min_{freq} = 0.02$  für die Klasse „verstorben“ (Klasse  $v$ ). Im dritten Versuch wurde diese Schwelle verändert zu  $\min_{freq} = 0.40$  bzw.  $\min_{freq} = 0.04$ , um eine Vergleichbarkeit der Ergebnisse zu gewährleisten. Die insgesamt erzeugten Regeln wurden nach diesen Kriterien gefiltert. Für die Testmenge wurde die minimale Konfidenz auf  $\min_{conf} = 0.99$  für die Klasse „überlebt“ und  $\min_{conf} = 0.40$  für die Klasse „verstorben“ gesetzt.

In der Tab. 6.1 sind die Ergebnisse sowie der Mittelwert aller drei Versuche zum Vergleich wiedergegeben.

Möchte man eine Regelmenge erzeugen, deren Regelanzahl deutlich unter der von 362 liegt, so kann ein recht großer Anteil der Daten nicht klassifiziert werden. Das wäre für eine Alarmgebung nicht so dramatisch; für diese Daten würde kein Alarm gegeben. Die Überlappung der Klassen der symbolischen Daten ist also ähnlich wie bei den metrischen Daten recht groß.

Ein Dilemma ist, dass es für die Klasse „überlebt“ sehr viele Regeln gibt, die eine sehr hohe Konfidenz nahe 1.00 (entspricht 100%) und eine recht große Häufigkeit besitzen, während die konfidenteren Regeln für die bedeutendere Klasse „verstorben“ eher eine niedrige Häufigkeit besitzen, was sich auch in der Auswahl der Schwellwerte widerspiegelt. Sehr interessant ist, dass die Frühzeitigkeit der Regeln der verstorbenen Patienten im Mittel deutlich höher ist als die der überlebenden.

	Versuch 1	Versuch 2	Versuch 3	Mittelwert
Anzahl der Regeln	21	26	22	23.0
Regeln (Kl. $\ddot{u}$ )	8	11	6	8.3
Regeln (Kl. $v$ )	13	15	16	14.7
Länge der Regeln (Kl. $\ddot{u}$ )	12.6	11.2	10.3	11.4
Länge der Regeln (Kl. $v$ )	15.0	10.9	16.1	14.0
korrekt klassifiziert [%]	51.38	58.01	53.04	54.14
falsch klassifiziert [%]	9.94	11.60	11.05	10.86
nicht klassifiziert [%]	38.67	30.39	35.91	34.99
Spezifität [%]	86.14	85.59	84.91	85.55
Sensitivität [%]	60.00	66.67	60.00	62.22
Test-Konfidenz (Kl. $\ddot{u}$ ) [%]	100.00	100.00	100.00	100.00
Test-Konfidenz (Kl. $v$ ) [%]	76.49	53.56	63.27	64.44
gew. Test-Konf. (Kl. $\ddot{u}$ ) [%]	100.00	100.00	100.00	100.00
gew. Test-Konf. (Kl. $v$ ) [%]	91.95	85.93	89.40	89.09
Test-Häufigkeit (Kl. $\ddot{u}$ ) [%]	35.77	35.46	41.34	37.52
Test-Häufigkeit (Kl. $v$ ) [%]	4.63	5.52	6.46	5.54
Frühzeitigkeit (Kl. $\ddot{u}$ ) [Tage]	4.07	4.27	4.47	4.27
Frühzeitigkeit (Kl. $v$ ) [Tage]	9.53	15.77	11.15	12.15

**Tabelle 6.1:** ASK-Daten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln. Die Sensitivität und die Spezifität wurde ohne die nicht klassifizierten Itemsets berechnet. Die Test-Konfidenz, die gewichtete (Abk.: gew.) Test-Konfidenz, die Häufigkeit und die Frühzeitigkeit ist pro Versuch als Mittel über die Anzahl der erzeugten Regeln angegeben. Bei der (gew.) Test-Konfidenz wurden nur Werte größer als die im Text genannten Schwellen für den Mittelwert berücksichtigt, da das Itemset sonst keine Regel dieser Klasse darstellt. Regeln der Klasse  $c$  (Zeile 2 und 3) sind Regeln, deren (gew.) Test-Konfidenz für diese Klasse maximal ist. Dasselbe gilt für die Länge der Regeln, die gemittelt pro Versuch angegeben ist.

Die mittlere Länge der Regeln ist mit 11.4 bzw. 14.0 deutlich kürzer als die maximal mögliche Länge von 48, aber dennoch beeinträchtigt die Länge der Regeln deren Verständlichkeit.

Anschließend werden in den Tabellen 6.2 und 6.3 die Ergebnisse der Wichtigkeit wiedergegeben. Die Werte der Wichtigkeiten wurden pro Klasse aus den 3 Versuchen addiert. Die Nummerierungen entsprechen den 48 Variablen und deren Negationen, die im Abschnitt A.4 beschrieben sind.

Nur 60 der 96 Items lieferten nach der Filterung der Regeln anhand der vorgegebenen Schwellen einen Beitrag (Wichtigkeit  $> 0$  für mind. eine Klasse) zu einer der drei Regelmengen. Von den wichtigeren Items (fett markiert in den Tabellen 6.2 und 6.3) sind 19 von 20 *verneinende* Items für die Klasse „überlebt“ (d.h. Wichtigkeit für diese Klasse  $> 0.19$ ) und 9 von 15 *bejahende* Items für die Klasse „verstorben“. Interessant ist also, dass verneinende Items ihre Berechtigung bei der Analyse der septischer Schock-Daten haben. Hinzu kommt, dass es auch medizinisch sinnvoll erscheint, dass das Nichtvorhandensein von Attributen (verneinende Items) eher der Klasse „überlebt“ entspricht, während das Vorhandensein von Attributen (bejahende Items) eher der Klasse „verstorben“ entspricht, wenngleich im letzteren Fall das nicht signifikant ist (9 von 15).

Die wichtigen Items 4, 23 und 76 sind für Regeln *beider* Klassen besonders wichtig. Wie kann das sein? Das ist nur möglich, wenn diese Items *zusammen* mit anderen Items wichtige Regeln bilden. Besonders interessant sind diejenigen Items, die nur für eine Klasse eine hohe Wichtigkeit haben.

Noch interessanter sind Variablen, deren Bejahung für eine Klasse wichtig ist und deren Verneinung für die andere Klasse wichtig ist. In diesem Fall charakterisiert die Variable beide Klassen, nicht nur eine. Die einzige Variable, für die deutlich beide Itemausprägungen vorkommen, ist die Hämofiltration (Items 39 und 40), der somit eine ausgezeichnete Bedeutung zukommt. Die einzige Variable, die in den Tabellen 6.2 und 6.3 weder eine Wichtigkeit für die eine noch für die andere Klasse hat, ist die Reoperation (Items 93 und 94). Die Reoperation kann man damit als unwichtigste Variable ansehen, zumindestens in diesem Datensatz.

D.h. aber nicht, dass die Reoperation medizinisch unwichtig ist; das Gegenteil ist der Fall. Die Aussage ist, dass innerhalb des ASK-Datensatzes die Reoperation nicht als wichtige Variable für die betrachtete Klassifikationsaufgabe charakterisiert wird.

Item-Nr.	Item-Name	überlebt	verstorben
1	<b>Organinsuffizienz</b>	0.0080	0.3234
2	not 1	0.0001	0.0036
4	<b>not Traumatisch</b>	0.5911	0.7132
5	GCS = 15	0.0034	0.1179
7	Metastatischer Tumor	0.0001	0.0036
8	not 7	0.0012	0.0360
10	not Bösartiges Haematom	0.0014	0.0428
12	not Diabetes mellitus präoperativ	0.0007	0.0166
14	not Diabetes mellitus insulinpflichtig	0.0007	0.0166
15	Leberzirrhose präoperativ	0.0001	0.0018
16	not 15	0.0008	0.0298
17	Erythrozytengabe	0.0015	0.1232
19	Plasmagabe	0.0038	0.0790
21	Thrombozytengabe	0.0001	0.0018
22	<b>not 21</b>	0.2169	0.0024
23	<b>SOFA (Score) &gt; 12</b>	1.0020	0.8072
25	<b>Multiorgan-Score &gt; 3</b>	0.0159	0.8072
27	<b>Anzahl der Organausfälle &gt; 2</b>	0.0159	0.8072
29	Parentale Ernährung	0.0026	0.0938
32	not Enterale Ernährung	0.0002	0.0292
33	Steroide	0.0001	0.0051
35	<b>Beatmung</b>	0.0056	0.3976
37	PAK	0.0012	0.0276
38	<b>not 37</b>	0.2267	0.0674
39	<b>Hämofiltration</b>	0.0048	0.4765
40	<b>not 39</b>	0.3122	0.0115
41	Dialyse	0.0004	0.0172
42	<b>not 41</b>	0.4064	0.2999
43	Peritonealspülung	0.0043	0.0991
44	not 43	0.0015	0.0953
46	<b>not Reanimation</b>	0.4485	0.1953
48	<b>not Cardioversion</b>	0.5289	0.1953

**Tabelle 6.2:** ASK-Daten (Items 1 bis 48): Summierte Wichtigkeit der Items (additiv aus den drei Versuchen gebildet). Die Items mit den höchsten zugehörigen Werten (für mind. eine Klasse  $> 0.19$ ) sind fett markiert. Die Werte für die nicht genannten Item-Nummern sind 0.

Item-Nr.	Item-Name	überlebt	verstorben
50	<b>not Schrittmacher</b>	0.8143	0.0615
51	Antiarrythmika-Gabe	0	0.0203
53	<b>Volumengabe um RR &gt; 100</b>	0.0033	0.1953
55	Hypotension trotz Substitution	0.0010	0.0944
57	Bronchoskopie	0.0010	0.1155
59	Relaxierung	0.0019	0.0593
62	<b>not Endoskopie</b>	0.9862	0.0024
64	<b>not Volle enterale Ernährung</b>	0.0057	0.1947
65	<b>Auffälligkeit Gastrointestinaltrakt</b>	0.0066	0.2308
68	<b>not Cholecystitis</b>	0.2010	0.0888
70	<b>not Stressulkus</b>	0.5755	0.0134
72	<b>not nekrotisierende Enterocolitis</b>	0.8635	0.1293
74	<b>not nekrotisierende Pankreatitis</b>	0.7154	0.0260
76	<b>not spontane Gallenblasenperf.</b>	0.8657	0.2308
77	akutes Nierenversagen	0.0001	0.0115
78	<b>not 77</b>	0.3662	0
79	Hämodialyse	0	0.0056
80	<b>not 79</b>	0.4861	0.0024
82	<b>not hämorrh. Diathese</b>	0.5427	0
83	mind. 3 Antibiotika gegeben	0.0001	0.0080
84	<b>not 83</b>	0.2128	0
85	Infektion	0.0010	0.0944
87	Eiter (abgeschlossener Raum)	0.0001	0.0120
88	<b>not 87</b>	0.4485	0.0065
89	Infektionsquelle nachgewiesen	0.0010	0.0944
90	not 89	0.1184	0
92	not Druckinfusion	0.1512	0.0128
95	<b>Katecholamingabe</b>	0.0126	0.7717

**Tabelle 6.3:** ASK-Daten (Items 49 bis 96): Summierte Wichtigkeit der Items (additiv aus den drei Versuchen gebildet). Die Items mit den höchsten zugehörigen Werten (für mind. eine Klasse > 0.19) sind fett markiert. Die Werte für die nicht genannten Item-Nummern sind 0.

Ein Beispiel für eine „verstorben“-Regel (aus dem 1. Versuch) ist:

„**if** Organinsuffizienz **and** not Traumatisch **and** SOFA > 12 **and** Multiorgan-Score > 3 **and** Anzahl der Organausfälle > 2 **and** Hämofiltration **and** Katecholamingabe **then** verstorben **with**  $wconf = 98.31\%$  **and**  $freq = 7.18\%$  **and** earliness = 12.2 [Tage].“

Ein Beispiel für eine „überlebt“-Regel ist:

„**if** SOFA > 12 **and** not Reanimation **and** not Cardioversion **and** not Endoskopie **and** not Stressulkus **and** not nekrotisierende Enterocolitis **and** not spontane Gallenblasenperforation **and** not hämorr. Diathese **and** not Eiter **and** not Infektionsquelle nachgewiesen **then** überlebt **with**  $wconf = 100.00\%$  **and**  $freq = 35.36\%$  **and** earliness = 3.6 [Tage].“

Man sieht anhand des Beispiels noch einmal deutlich die Dominanz der Bejahungen bei „verstorben“-Regeln und die Dominanz der Verneinungen bei „überlebt“-Regeln. Interessanterweise sieht man auch, dass der SOFA-Score > 12 allein *nicht* den Ausschlag für „überlebt“ oder „verstorben“ gegeben hat.

Da sehr viele Items (36 von 96) keinen Beitrag für diese Regeln lieferten, viele Items nur eine geringe Wichtigkeit haben und es auch einige Überschneidungen in der Bedeutung der Variablen gibt (Organinsuffizienz, SOFA > 12, Multiorgan-Score > 3, Anzahl der Organausfälle > 2), wird im nächsten Abschnitt unter Berücksichtigung dieser Punkte mit den in den Tabellen 6.2 und 6.3 fett markierten Itemsets der Versuch wiederholt.

## 6.3 Generalisierung unter Berücksichtigung der Wichtigkeit

Der folgende Versuch verwendet 27 der 29 in den Tabellen 6.2 und 6.3 fett markierten Items aus den insgesamt 96 vorhandenen. Die Items 23 und 25 wurden wegen medizinischer Überschneidung mit dem Item 27 weggelassen. Überprüft werden soll, ob man mit den ausgewählten „wichtigen“ Items vergleichbare Ergebnisse wie im letzten Abschnitt erhält.

Ausgehend von den 362 Patienten, deren Variablenwerte als Itemsets kodiert wurden, wurden wiederum drei Versuchswiederholungen durchgeführt. Die Heuristiken 1., 2. und 4. aus dem Abschnitt 4.6 wurden angewendet. Für das erste Level wurden 50% der Daten zum Training verwendet, die anderen 50% zum Testen, u.a zur Bestimmung der Klassifikationsleistung. Es lagen jeweils folgende Einstellungen vor:  $maxlevel = 5$ ,  $max_{new} = 3$ ,  $min_{freq} = 0.20$  (2. Versuch: 0.14, 3. Versuch: 0.16) sowohl

	Versuch 1	Versuch 2	Versuch 3	Mittelwert
Anzahl der Regeln	25	30	29	28.0
Regeln (Kl. ü)	14	19	21	18.0
Regeln (Kl. v)	11	11	8	10.0
Länge der Regeln (Kl. ü)	12.3	14.0	12.0	12.8
Länge der Regeln (Kl. v)	11.9	13.3	9.6	11.6
korrekt klassifiziert [%]	74.03	64.09	65.75	67.96
falsch klassifiziert [%]	13.26	18.79	23.20	18.42
nicht klassifiziert [%]	12.71	17.13	11.05	13.63
Spezifität [%]	87.88	80.16	74.63	80.89
Sensitivität [%]	72.00	62.50	70.37	68.29
Test-Konfidenz (Kl. ü) [%]	96.23	96.88	96.98	96.70
Test-Konfidenz (Kl. v) [%]	47.58	27.67	49.46	41.57
gew. Test-Konf. (Kl. ü) [%]	81.89	84.39	85.22	83.83
gew. Test-Konf. (Kl. v) [%]	81.61	69.89	84.59	78.70
Test-Häufigkeit (Kl. ü) [%]	33.43	19.45	28.10	27.00
Test-Häufigkeit (Kl. v) [%]	11.75	16.17	9.39	12.44
Frühzeitigkeit (Kl. ü) [Tage]	5.57	4.74	5.03	5.11
Frühzeitigkeit (Kl. v) [Tage]	9.95	11.10	12.08	11.04

**Tabelle 6.4:** ASK-Daten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln unter Berücksichtigung der Wichtigkeit. Die weiteren Erklärungen sind identisch zu denen in der Tab. 6.1.

für die Trainings- als auch für die Test-Itemsets für die Klasse „überlebt“ (Klasse ü) und  $\min_{freq} = 0.02$  (alle Versuche) für die Klasse „verstorben“ (Klasse v). Für die Testmenge wurde die minimale Konfidenz auf  $\min_{conf} = 0.92$  (3. Versuch: 0.90) für die Klasse „überlebt“ und  $\min_{conf} = 0.22$  (2. Versuch: 0.18, 3. Versuch: 0.25) für die Klasse „verstorben“ gesetzt. Die erzeugten Regeln wurden nach diesen Kriterien gefiltert.

In der Tab. 6.4 sind die Ergebnisse sowie der Mittelwert aller drei Versuche zum Vergleich wiedergegeben.

**Fazit:** Ein erster Vorteil der Analyse mit 27 statt 96 Items ist die Reduzierung der Laufzeit um einen Faktor  $> 10$ . Die Anzahl der Regeln ist etwas höher, allerdings sind nun mehr Regeln für die Klasse „verstorben“ erzeugt worden, was daran liegen kann, dass die Indikatoren für diese Klasse deutlicher hervorgetreten sind. Die Länge der Regeln ist etwa gleich. Etwa 10% der Daten sind anstelle einer Nicht-Klassifikation nun korrekt klassifiziert worden, allerdings ohne deutliche Veränderung der Sensitivität oder Spezifität, wenngleich die Klasse „verstorben“ etwas profitiert hat. Die gewichtete und ungewichtete Test-Konfidenz sowie die Test-Häufigkeit

für die Klasse „überlebt“ ist etwas geringer geworden. Die Test-Häufigkeit für die Klasse „verstorben“ ist etwas höher. Die Frühzeitigkeit für die Klasse „verstorben“ ist im Mittel wieder etwa 6 Tage höher als die der Klasse „überlebt“. Insgesamt ist es also durchaus möglich mit einer Auswahl von Items (hier 27 von 96, Reduktionsfaktor: 3.6) ähnlich gute Ergebnisse zu erzielen wie mit allen Items; Abweichungen treten auf, da sicherlich das eine oder andere hier nicht verwendete Item in den Versuchen des letzten Abschnitts Bestandteil einer konfidenten Regel war. Positiv ist zu sehen, dass die Klassifikationsleistung besser wurde; allerdings wurden auch 5 Regeln mehr erzeugt, so dass nicht eindeutig zu entscheiden ist, ob die bessere Klassifikation dadurch verursacht wird. Auch kann die Wahl der Schwellwerte eine Rolle spielen.

Betrachtet man abschließend die Wichtigkeit der in den Regeln tatsächlich aufgetretenen Items, so stellt man fest, dass von den 27 Items tatsächlich nur 26 auftreten, siehe Tab. 6.5. Das Item mit der Nr. 68 (not Cholecystitis) tritt nicht auf.

Während die verneinenden Attribute besonders wichtig für die Klasse „überlebt“ sind, sind die bejahenden Attribute eher besonders wichtig für die Klasse „verstorben“. Die Attribute 4 und 53 lieferten hier für beide Klassen eine ähnlich hohe Wichtigkeit.

Eine kurze und sichere „verstorben“-Regel (aus dem 1. Versuch) ist:

**„if not Traumatisch and Anzahl der Organausfälle > 2 and Beatmung and Hämo-  
filtration and not Dialyse and Katecholamingabe then verstorben with  $wconf = 89.73\%$  and  $freq = 6.63\%$  and earliness = 17.0 [Tage].“**

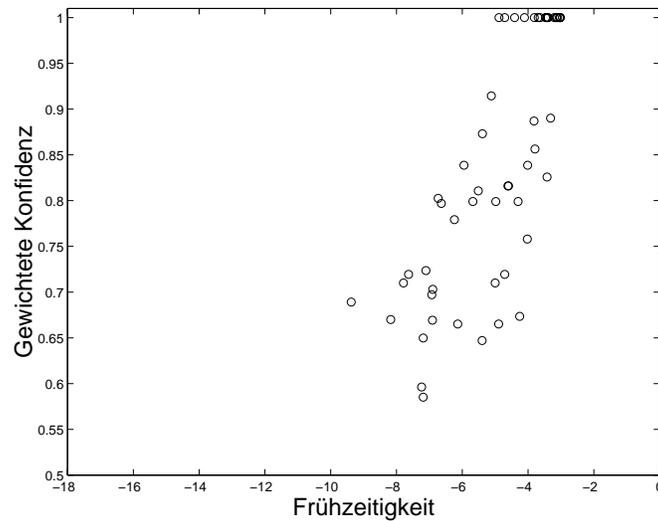
Ein Beispiel für eine „überlebt“-Regel ist:

**„if not Thrombozytengabe and not PAK and not Hämofiltration and not Dialyse and not Reanimation and not Cardioversion and not Schrittmacher and not Endoskopie and not Volle enterale Ernährung and not Stressulkus and not nekrotisierende Enterocolitis and not nekrotisierende Pankreatitis and not spontane Gallenblasenperforation and not Hämodialyse and not hämorrh. Diathese and not mind. 3 Antibiotika gegeben and not Eiter (abgeschlossener Raum) then überlebt with  $wconf = 100.00\%$  and  $freq = 26.52\%$  and earliness = 3.5 [Tage].“**

Trotz des vielversprechenden Ansatzes, kann auch hier das generelle medizinische Problem der Invidualität und der Überlappung der Patientendaten nicht gelöst werden, da für eine bessere Generalisierungsleistung eine höhere Fallzahl notwendig wäre. Die nicht klassifizierten Fälle sind entweder Einzelfälle (zu niedrige Häufigkeit) oder Fälle, in denen keine eindeutige Entscheidung möglich ist (zu niedrige Konfidenz).

Item-Nr.	Item-Name	überlebt	verstorben
1	Organinsuffizienz	0.0989	0.5321
4	not Traumatisch	1.0654	1.1332
22	not Thrombozytengabe	0.5020	0.0422
27	Anzahl der Organausfälle > 2	0.7318	1.1364
35	Beatmung	0.1482	0.8064
38	not PAK	0.8350	0.3278
39	Hämofiltration	0.0082	0.4370
40	not 39	0.9446	0.1482
42	not Dialyse	1.1560	0.5973
46	not Reanimation	0.6607	0.1831
48	not Cardioversion	0.7187	0.1856
50	not Schrittmacher	0.7347	0.1484
53	Volumengabe um RR > 100	0.1363	0.1625
62	not Endoskopie	0.5717	0.0299
64	not Volle enterale Ernährung	0.3625	0.0346
65	Auffälligkeit Gastrointestinaltrakt	0.2724	0.1701
70	not Stressulkus	0.6151	0.1250
72	not nekrotisierende Enterocolitis	0.7137	0.1097
74	not nekrotisierende Pankreatitis	0.5681	0.1156
76	not spontane Gallenblasenperforation	0.7361	0.1859
78	not akutes Nierenversagen	0.3155	0.0095
80	not Hämodialyse	0.4793	0.0229
82	not hämorrh. Diathese	0.6023	0.0260
84	not mind. 3 Antibiotika gegeben	0.3138	0.0134
88	not Eiter (abgeschlossener Raum)	0.3214	0.0129
95	Katecholamingabe	0.4366	1.0972

**Tabelle 6.5:** ASK-Daten: Summierte Wichtigkeit der „wichtigsten“ Items (additiv aus den drei Versuchen gebildet). Die Werte für das einzige nicht genannte Item 68 sind 0.



**Abbildung 6.1:** ASK-Daten: Gewichtete Konfidenz in Abhängigkeit von der Frühzeitigkeit für die Klasse „überlebt“ (Kreis).  $-x$  [Tage] bedeutet  $x$  Tage vor der Entlassung bzw. vor dem Versterben.

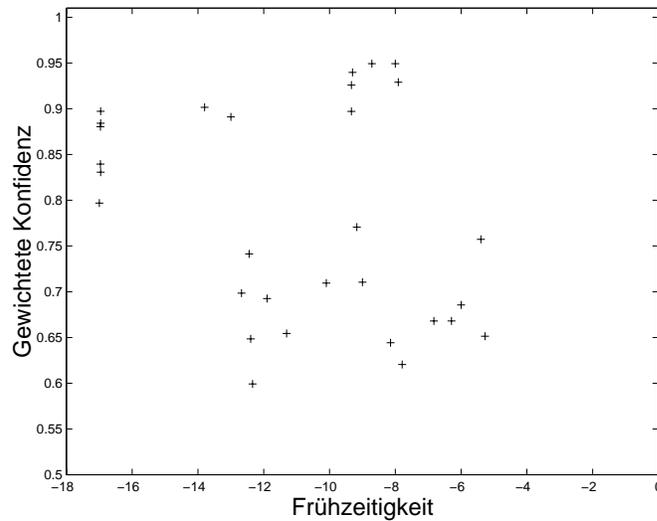
Einen interessanten Einblick in die Struktur der generierten Regeln bez. der Frühzeitigkeit geben die Abbildungen 6.1, 6.2 und 6.3. Dort sind die gewichteten Konfidenzen der Regeln in Abhängigkeit von der Frühzeitigkeit angegeben, einmal für die Regeln der Klasse „überlebt“, einmal für die der Klasse „verstorben“ und einmal für die beider Klassen gemeinsam. Es wurden alle Regeln aus den Regelmengen der drei Versuche jeweils übereinander abgebildet.

In der Abb. 6.3 kann man eine Einteilung in drei Gruppen vornehmen:

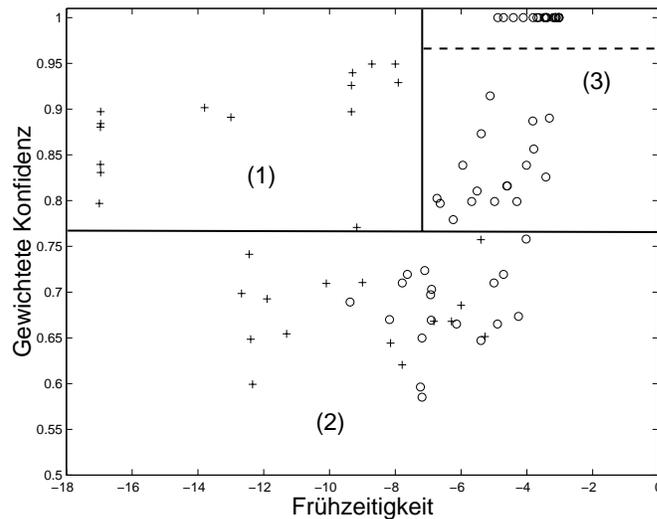
1. Frühwarnungen mit hoher gewichteter Konfidenz für „verstorben“,
2. gemischte Warnungen, die eine besondere Aufmerksamkeit des Arztes erfordern, da bei Eintreten dieser eher nicht so sicheren Regeln sich noch entscheiden wird, ob der Patient verstirbt oder überlebt. Hier kann eine erhöhte Aufmerksamkeit des Arztes evtl. die Überlebenschance des Patienten verbessern („Aufmerksamkeitsregeln“) und
3. Entwarnungen für „überlebt“, die oberhalb der gestrichelten Linie besonders sicher (gew. Konfidenz = 1) sind.

## 6.4 Generalisierung und Ähnlichkeit

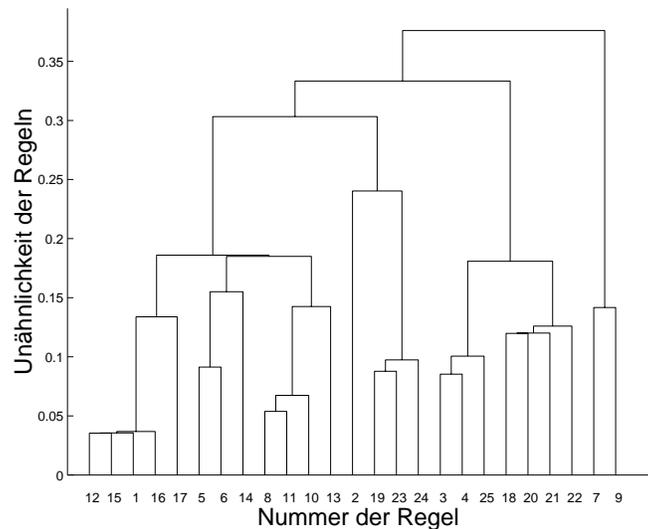
Abschließend wird an die Abschnitte 4.9 und 6.3 angeknüpft, wo *ähnliche* Regeln und Regelmengen betrachtet wurden. Im Sinne der dort genannten Definitionen wä-



**Abbildung 6.2:** ASK-Daten: Gewichtete Konfidenz in Abhängigkeit von der Frühzeitigkeit für die Klasse „verstorben“ (Kreuz).  $-x$  [Tage] bedeutet  $x$  Tage vor der Entlassung bzw. vor dem Versterben.



**Abbildung 6.3:** ASK-Daten: Gewichtete Konfidenz in Abhängigkeit von der Frühzeitigkeit für beide Klassen gemeinsam.  $-x$  [Tage] bedeutet  $x$  Tage vor der Entlassung bzw. vor dem Versterben. Es sind drei Gruppen erkennbar: (1) Frühwarnungen für „verstorben“ (Kreuz), (2) gemischte Warnungen, die eine besondere Aufmerksamkeit des Arztes erfordern und (3) Entwarnungen für „überlebt“ (Kreis). Die besonders sicheren Entwarnungen liegen oberhalb der gestrichelten Linie.



**Abbildung 6.4:** ASK-Daten: Dendrogramm für die Unähnlichkeit der erzeugten Generalisierungsregeln.

re der Versuch 1 der repräsentative (Parametereinstellungen:  $\tau = 0.2$ ,  $\kappa_1 = 0.2$  für die Häufigkeit auf den Testdaten,  $\kappa_2 = \kappa_3 = 0.2$  für die Konfidenz auf den Testdaten für beide Klassen) mit  $Sim(R_1) = 0.7462$ . Für die beiden anderen Regelmengen galt  $Sim(R_3) = 0.7425$  und  $Sim(R_2) = 0.7341$ . Da hier nur drei Regelmengen vorliegen und die Werte nahe beieinander liegen, geschieht die Auszeichnung einer Regelmengen hier nur zu Demonstrationszwecken.

Ausgehend von der Regelmenge  $R$  des 1. Versuchs wird nun aus der berechneten Ähnlichkeitsmatrix  $S$  (Ähnlichkeit aller Regeln aus  $R$  zueinander) eine Unähnlichkeitsmatrix  $U$  berechnet, indem die Diagonale auf Null gesetzt wird und  $U_{ik} := 1 - S_{ik}$  berechnet wird; dann ist  $U = U^T$ . Eine Clusteranalyse mit dem single-linkage-Verfahren [Bac96, Kap. 3] liefert das in der Abb. 6.4 gezeigte Dendrogramm.

Man kann anhand des Dendrogramms z.B. folgende Regeln clustern:

1. 12, 15, 1, 16 (100%-ige gew. Testkonfidenz bez. der Klasse „überlebt“),
2. 17 (sehr hohe gew. Testkonfidenz bez. der Klasse „überlebt“),
3. 5, 6 (hohe gew. Testkonfidenz bez. der Klasse „überlebt“),
4. 4 (hohe gew. Testkonfidenz bez. der Klasse „überlebt“),
5. 7, 9 (hohe gew. Testkonfidenz bez. der Klasse „überlebt“),
6. 8, 11, 10 (hohe gew. Testkonfidenz bez. der Klasse „überlebt“),

7. 13 (hohe gew. Testkonfidenz bez. der Klasse „überlebt“),
8. 3, 4, 25 (sehr hohe gew. Testkonfidenz bez. der Klasse „verstorben“),
9. 19, 23 (sehr hohe gew. Testkonfidenz bez. der Klasse „verstorben“),
10. 24 (hohe gew. Testkonfidenz bez. der Klasse „verstorben“),
11. 18, 20, 21, 22 (hohe gew. Testkonfidenz bez. der Klasse „verstorben“),
12. 2 (hohe gew. Testkonfidenz bez. der Klasse „verstorben“).

Präsentiert man Medizinexperten Regeln, so bietet es sich an, zuerst Regeln aus den 12 Clustern, also 12 Regeln zu präsentieren. Damit hat man die wichtigsten untereinander unähnlichen Regeln abgedeckt. Im Vergleich zu einer Präsentation aller 25, evtl. sehr ähnlichen Regeln, ist das eine Ersparnis von etwa 50%, was eine fundiertere redundanzfreie Diskussion ermöglicht.

## 6.5 Generalisierung und Generalisierungsindices

In diesem Abschnitt wird im Anschluss an den Abschnitt 4.7 der innere Generalisierungsindex  $G^{(in)}$  und der äußere Generalisierungsindex  $G^{(out)}$  bei der Generalisierung verwendet, um die Regelgenerierung zu beschleunigen. Generalisiert wird der Datensatz, der auch schon im Abschnitt 6.3 generalisiert wurde (362 Patienten, 27 Items).

Ausgehend von den 362 Patienten, deren Variablenwerte als Itemsets kodiert wurden, wurden auch hier erneut drei Versuchswiederholungen durchgeführt. Es wurden diesmal nur die Heuristiken 1. und 4. aus dem Abschnitt 4.6 angewendet, d.h. auf eine maximale Levelzahl (2.) wurde zugunsten des äußeren Generalisierungsindex verzichtet. Für das erste Level wurden 50% der Daten zum Training verwendet, die anderen 50% zum Testen, u.a zur Bestimmung der Klassifikationsleistung. Es lagen jeweils folgende Einstellungen vor:  $\max_{new} = 3$ ,  $\min_{freq} = 0.25$  (3. Versuch: 0.18) sowohl für die Trainings- als auch für die Test-Itemsets für die Klasse „überlebt“ (Klasse  $\bar{u}$ ) und  $\min_{freq} = 0.02$  (alle Versuche) für die Klasse „verstorben“ (Klasse  $v$ ). Für die Testmenge wurde die minimale Konfidenz auf  $\min_{conf} = 0.90$  (3. Versuch: 0.92) für die Klasse „überlebt“ und  $\min_{conf} = 0.22$  für die Klasse „verstorben“ gesetzt. Die erzeugten Regeln wurden nach diesen Kriterien gefiltert.

Die Fenstergröße zur Berechnung von  $G^{(in)}$  wurde auf 15, die Abbruchschwelle für das Unterschreiten von  $G^{(in)}$  wurde auf 0.4 und die Abbruchschwelle für das Unterschreiten von  $G^{(out)}$  wurde auf 0.15 gesetzt.

**Zur Zeitersparnis:** Das Verfahren benötigte in den 3 Versuchen 5, 1 bzw. 4 Minuten Laufzeit, im Mittel also 3 Minuten. Die Generalisierung im vorherigen Abschnitt 6.3 (also ohne Generalisierungsindices) benötigte 174, 179 bzw. 58 Minuten, im Mittel 137 Minuten. Die Generalisierungsindices brachten damit eine Verbesserung der Laufzeit um den Faktor 45, das sind nur 2.2% der 137 Minuten. Das Verfahren wird

	Versuch 1	Versuch 2	Versuch 3	Mittelwert
Anzahl der Level	4	3	5	4.0
Anzahl der Regeln	22	24	27	24.3
Regeln (Kl. ü)	8	13	14	11.7
Regeln (Kl. v)	14	11	13	12.7
Länge der Regeln (Kl. ü)	12.9	11.2	10.6	11.6
Länge der Regeln (Kl. v)	12.0	10.5	10.2	10.9
korrekt klassifiziert [%]	75.14	76.80	62.98	71.64
falsch klassifiziert [%]	17.13	13.26	17.68	16.02
nicht klassifiziert [%]	7.73	9.94	19.34	12.34
Spezifität [%]	82.88	88.32	79.20	83.47
Sensitivität [%]	71.43	69.23	71.43	70.70
Test-Konfidenz (Kl. ü) [%]	95.16	94.48	97.61	95.75
Test-Konfidenz (Kl. v) [%]	35.66	47.58	35.64	39.63
gew. Test-Konf. (Kl. ü) [%]	82.47	74.38	89.64	82.16
gew. Test-Konf. (Kl. v) [%]	69.92	81.61	71.39	74.31
Test-Häufigkeit (Kl. ü) [%]	45.93	40.42	25.77	37.37
Test-Häufigkeit (Kl. v) [%]	15.83	11.75	17.43	15.00
Frühzeitigkeit (Kl. ü) [Tage]	5.26	6.30	5.23	5.60
Frühzeitigkeit (Kl. v) [Tage]	10.83	9.95	10.55	10.46

**Tabelle 6.6:** ASK-Daten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln unter Berücksichtigung des inneren und äußeren Generalisierungsindex. Die weiteren Erklärungen sind identisch zu denen der Tab. 6.1.

damit von der Laufzeit gesehen gut anwendbar, wobei vernünftige Schwellen angenommen werden müssen, z.B. wie oben gewählt.

In der Tab. 6.6 sind die Ergebnisse sowie der Mittelwert aller drei Versuche zum Vergleich wiedergegeben.

Die Ergebnisse mit und ohne Generalisierungsindices (vgl. Tab. 6.4 und 6.6) sind in allen Punkten vergleichbar gut, so dass die Generalisierungsindices bei gleicher Güte einen deutlichen Laufzeitvorteil einbrachten.

Berechnet man den durchschnittlichen Flächeninhalt der ROC-Kurven jeweils für die drei dargestellten Klassifikatoren aus den Abschnitten 6.2, 6.3 und 6.5, so erhält man die Werte, die in der Tab. 6.7 wiedergegeben sind. Der Wert von 0.824 für die Klassifikatoren aus dem Abschnitt 6.2 relativiert sich, wenn man gezwungen ist, alle Samples zu klassifizieren, da in diesen Versuchen recht viele Samples als nicht-klassifizierbar eingestuft wurden. Insgesamt ist das Ergebnis aus dem Abschnitt 6.5 am besten, da für alle drei Zuordnungen die besten Werte erzielt wurden.

Klassifikatoren	$F, (1)$	$F, (2)$	$F, (3)$
Abschnitt 6.2	0.824	0.629	0.710
Abschnitt 6.3	0.803	0.724	0.729
Abschnitt 6.5	0.828	0.740	0.759

**Tabelle 6.7:** ASK-Daten: Mittlere Flächeninhalte  $F$  der ROC-Kurven. (1) Sensitivität und Spezifität wurden ohne die nicht-klassifizierten Samples berechnet, (2) die nicht-klassifizierten Samples wurden der Klasse „überlebt“ zugeordnet, (3) die nicht-klassifizierten Samples wurden der Klasse „verstorben“ zugeordnet. In den Fällen (2) und (3) wurden Sensitivität und Spezifität mit allen Samples berechnet.

Man kann sagen, dass sich die Anwendung der Wichtigkeit, d.h. die Auswahl der wichtigsten Items, und die Anwendung der vorgestellten Heuristiken lohnt. Anhand der berechneten Flächeninhalte kann man das Dilemma der Klassifikation der (symbolischen) septischer Schock-Daten erkennen: Möchte man einen hohen Wert für den Flächeninhalt unter der ROC-Kurve haben, der für eine gute Klassifikationsleistung steht, so ist man gezwungen, Samples (die ja hier Patienten entsprechen) als nicht-klassifizierbar (Einzelfälle) einzustufen. Möchte man alle Patienten klassifizieren, so sinkt die Klassifikationsgüte.

**Fazit:** Im Vergleich zu der Zustandsklassifikation der metrischen Patientensamples (jedem Patienten konnten mehrere Samples zugeordnet sein) auf den Vorstudien-daten im letzten Kapitel lassen sich die symbolischen Daten der Patienten (jeder Patient entsprach genau einem Sample) besser klassifizieren (vgl. die Flächeninhalte unter den ROC-Kurven der Abb. B.3 und der Tab. 6.7). Zwei Aspekte machen diesen Unterschied aus: bessere Dokumentationsqualität der ASK-Daten und Klassifikation der Patienten (alle Daten pro Patient) bez. der ASK-Daten im Vergleich zur Klassifikation der Patientenzustände (Tageswerte) der Vorstudien-daten. Ob die symbolischen oder metrischen Daten sich generell besser zur Klassifikation von septischen Schock-Patienten eignen, kann aus den Ergebnissen nicht geschlossen werden. Im Kapitel 7, in dem sowohl die metrischen als auch die symbolischen Klinikdaten untersucht werden, werden für diese Fragestellung einige Antworten gegeben.

## 6.6 Vergleich mit Assoziationsregeln und FP-Bäumen

In Anwendungen wird häufig das A-priori-Verfahren [AS94] (vgl. Abschnitte 2.5.2 und 4.8) aufgrund der Einfachheit des Algorithmus verwendet. Ein aktueller Ansatz, Regeln auf der Basis von häufigen Mustern mit sog. FP-Bäumen zu generieren, wird in [HPY00] vorgestellt.

In [PB01] werden mit diesem Ansatz Regeln für die ASK-Daten generiert. Dort ist auch der Algorithmus beschrieben, der in Ergänzung zu [HPY00] auch eine Konfidenz-Berechnung durchführt. Diese Anwendung wird im Folgenden wiedergegeben. Anschließend werden mit dem A-priori-Algorithmus Regeln für die ASK-Daten generiert. Die Anzahl der erzeugten Regeln sowie die Laufzeit der Algorithmen wird am Ende des Abschnitts diskutiert und mit den Ergebnissen der (heuristischen) Generalisierungsregeln verglichen.

**Anwendung des FP-Baums:** Die Beispiel-Regeln aus [PB01] seien hier wiedergegeben (es wurden aus Laufzeitgründen nur 30 der 48 Attribute verwendet): Es sei  $\min_{freq} = 0.020$  und  $\min_{conf} = 0.750$  für die Klasse „verstorben“. Weil die überlebenden Patienten leichter mit Regeln zu beschreiben sind, sei  $\min_{freq} = 0.165$  und  $\min_{conf} = 0.980$  für die Klasse „überlebt“. 1284 Regeln für die Klasse „verstorben“ und 9976 Regeln für die Klasse „überlebt“, die häufig und konfident bez. der o.g. Schwellen sind, wurden erzeugt. Zwei der besten Regeln für überlebende und verstorbene Patienten seien hier genannt, wobei die Konfidenz- und Häufigkeitswerte auf *allen* Daten berechnet wurden (deskriptiver Ansatz), so dass die Ergebnisse nicht vollständig mit denen der Generalisierung (Analyse der Testdaten) vergleichbar sind.

1) „**if** Peritonealspülung = no **and** Thrombozytenkonzentrat = no **and** Haemodialyse = no **then** Klasse überlebt **with** confidence 0.98 **and** frequency 0.42“

2) “**if** Haemofiltration = no **and** Reoperation = no **and** akuter Nierenausfall = no **and** Leberzirrhose = no **then** Klasse überlebt **with** confidence 0.99 **and** frequency 0.29“

3) „**if** mind. 3 Antibiotika gegeben = yes **and** künstliche Beatmung = yes **and** enterale Ernährung = no **then** Klasse verstorben **with** confidence 0.82 **and** frequency 0.03“

4) „**if** Organausfall = yes **and** Antiarrythmika = yes **and** Haemodialyse = yes **and** Peritonealspülung = yes **then** Klasse verstorben **with** confidence 0.80 **and** frequency 0.03“

Die Laufzeit der FP-Baum-Regelgenerierung auf den ASK-Daten betrug 66 min. für den Baum der Klasse „überlebt“ und 138 min. für den der Klasse „verstorben“, also insgesamt 204 min. Dabei wird die Zeit durch die Generierung der Regeln bestimmt; die Konstruktion des initialen FP-Baums dauert jeweils etwa nur 1 s (alle Versuche jeweils Intel Pentium II 333MHz/256MB RAM, Windows NT 4.0).

**Generierung von Assoziationsregeln:** Verwendet wird eine im Anhang A.5 beschriebene Software. Auf eine Wiedergabe von Regelbeispielen wird verzichtet, da

Klasse	Konfidenz [%]	Häufigkeit [%]	Laufzeit	Anzahl Regeln
ü	100	30	16 s	919
ü	98	30	19 s	5916
ü	96	30	24 s	12955
ü	100	16.5	9 min.	280893
ü	98	16.5	10 min.	455622
ü	96	16.5	10 min.	605556
v	50	30	3 min.	0
v	30	30	4 min.	0
v	20	30	4 min.	474
v	10	30	5 min.	4861
v	0	30	12 min.	49899
v	75	30	1 min.	0
v	75	25	4 min.	0
v	75	20	11 min.	0
v	75	15	45 min.	0
v	75	10	3 h*	-
v	75	2	1 Tag*	-

**Tabelle 6.8:** ASK-Daten: Laufzeit und Anzahl der erzeugten Regeln mit dem A-priori-Algorithmus bei vorgegebener Konfidenz- und Häufigkeitsschwelle. „ü“ steht für die Klasse „überlebt“, „v“ für „verstorben“. Es wurde auf die jeweils angegebene Einheit gerundet. \* Die Laufzeiten für die beiden letzten Werte wurden geschätzt, daher kann keine Regelanzahl angegeben werden; bei diesen Einstellungen müssten sehr viele Kombinationen gebildet werden.

sie von der Struktur ähnlich wie die o.g. FP-Baum-Regeln sind. Interessanter ist für den anschließenden Vergleich die Anzahl der erzeugten Regeln und die Laufzeit. In der Tab. 6.8 werden für verschiedene Konfidenz- und Häufigkeitsschwellen die Ergebnisse wiedergegeben. Auf der Konklusionsseite der Regel durfte nur die Klasseninformation stehen, nicht etwa beliebige Attributkombinationen.

Das Fazit hat einen qualitativen Charakter, da die Implementierungen der Algorithmen in unterschiedlichen Sprachen vorgenommen wurden (A-priori in C, FP-Baum in JAVA, Generalisierungsregeln in MATLAB-C), so dass daraus unterschiedliche Ausführungsgeschwindigkeiten resultieren.

**Fazit:** Die Laufzeit des FP-Baums und des heuristischen Generalisierungsalgorithmus liegen mit etwa 2 bis 3h in der gleichen Größenordnung. Allerdings wurden bei den heuristischen Generalisierungsregeln sogar 96 Items berücksichtigt, dafür wurde aber nur auf den Trainingsdaten trainiert, bei etwas geringerer Laufzeit von

137 Minuten (siehe den Abschnitt 6.5), FP-Baum: 204 min. bei 60 Items. Bei 27 Items betrug die Laufzeit des heuristischen Generalisierungsalgorithmus nur 3 Minuten. Der Speicherplatzbedarf für die Regelgenerierung mittels FP-Baum ist mit ca. 500MB durch die rekursive Sub-FP-Baum-Erzeugung sehr hoch.

Die Laufzeit des Assoziationsregelmechanismus ist für die Klasse „überlebt“ im Minutenbereich, bei höheren Häufigkeitsschwellen sogar im Sekundenbereich. Für die Klasse „verstorben“ ergibt sich für den A-priori-Algorithmus ein sehr unbefriedigendes Bild: Da für diese Regeln eine Häufigkeitsschwelle von etwa 2% angenommen werden müsste, um überhaupt Regeln zu erhalten, explodiert die Laufzeit und der Speicherplatzbedarf dramatisch, so dass ohne eine intelligente Parallelverarbeitung dieses Verfahren **nicht** in der Lage ist, auf den ASK-Daten ein Ergebnis innerhalb eines sinnvollen Zeitrahmens zu produzieren.

Sehr unbefriedigend ist auch die Anzahl der erzeugten Regeln beim A-priori-Verfahren: Bei niedrigen Häufigkeitsschwellen werden sehr viele Regeln für die Klasse „überlebt“, aber keine Regeln für die Klasse „verstorben“ erzeugt, verbunden mit einem stark anwachsendem Speicher- und Laufzeitbedarf. Die Wahl der richtigen Schwellen ist somit kritischer als bei den anderen beiden Verfahren.

Grundsätzlich erzeugt der Generalisierungsregel-Mechanismus erst alle Regeln, die a-posteriori geprunt werden können, so dass die Wahl der Schwellen am unkritischsten ist. Ist die Häufigkeitsschwelle zu hoch, so muss bei den anderen beiden Verfahren die gesamte Regelerzeugung mit einer niedrigeren Schwelle wiederholt werden.

Der Generalisierungsregel-Mechanismus erlaubt direkt die Klassifikation anhand weniger Regeln und ist somit geeigneter für die Problemstellung der symbolischen Regelgenerierung mit integrierter Klassifikation. Da für eine Klassifikation mit den anderen beiden Verfahren auf der Grundlage der großen Menge an Regeln die redundanten eliminiert werden müssten, z.B. nach der Methode aus [LHM98], wird hier dieser Ansatz zur Klassifikation nicht weiter verfolgt.

Die einfache Klassifikation zusammen mit dem Vorteil der Vermeidung der Kontextverwischung (siehe Abschnitt 4.8) lässt die Methode der heuristischen Generalisierungsregeln im Vergleich zu den beiden anderen gängigen Methoden mindestens als gleichwertig erscheinen. Die „einfachste“ (und daher beliebteste) Methode, A-priori, erscheint am ungeeignetsten (für die ASK-Daten). Natürlich werden die Unterschiede bei kleineren Datensätzen mit weniger Attributen und deutlicherer Klassenstruktur nicht so stark oder kaum auffallen.

Der medizinische ASK-Datensatz mit einer bereits recht hohen Anzahl von Variablen und einer recht inhomogenen Patientenmenge hat bei allen drei Verfahren zu Laufzeiten im Stundenbereich geführt, die nicht für eine online-Datenbank-Abfrage

geeignet sind: Eine *Parallelisierung* von Datenanalyse-Algorithmen wird in Zukunft bei immer größeren und kompliziert strukturierten Datensätzen unter diesem Gesichtspunkt an Bedeutung gewinnen.

# Kapitel 7

## Die septischer Schock-Klinikdaten

Dieses Kapitel behandelt die Analyse der MEDAN-Klinikdaten, die bis Ende August 2001 deutschlandweit gesammelt wurden (vgl. Anhang A.3). Es handelt sich um 138 Patienten (68 überlebt, 70 verstorben), bei denen mindestens einmal während ihres dokumentierten Intensivaufenthaltes ein septischer Schock gesichert im Sinne der Def. 5.1.1 aufgetreten ist.

Im nächsten Abschnitt wird analog zu den Abschnitten 5.2 und 5.3 auf die Datenvorverarbeitung, auf grundlegende Verteilungseigenschaften und auf Korrelationsanalysen eingegangen. In den darauffolgenden Abschnitten werden umfangreiche Versuche zur metrischen, symbolischen und kombinierten Regelgenerierung auf der Grundlage der Klinikdaten durchgeführt. Im Abschnitt 7.5 werden dann auf der Basis der bis dahin erzielten Ergebnisse aus den metrischen Regelmengen Warnsysteme aufgebaut und deren Leistungsfähigkeit verglichen.

### 7.1 Vorverarbeitung und deskriptive Statistik

Die in diesem Abschnitt durchgeführte Datenvorverarbeitung wird als informative Übersicht wiedergegeben, da eine vollständige Darstellung von z.B. allein über 100 Histogrammen für die einzelnen Variablen den Rahmen sprengen würde. In der Medizin werden aber einige Doktorarbeiten angefertigt, die ausführlicher einzelne Teilaspekte der Epidemiologie beleuchten werden. Die Datenvorverarbeitung und die deskriptive Statistik umfassen Untersuchungen zu

- Grenzwertverletzungen,
- Füllgrad der Daten (in Abhängigkeit des Samplings),
- Anzahl der Patienten pro Variable,
- Zeitreihendynamik der Variablen,
- Verwendungsmöglichkeit der Variablen, Ausschluss von Variablen,

- Verteilungseigenschaften der metrischen Variablen, Behandlung fehlender Werte,
- Epidemiologie, insbesondere der Diagnosen und Operationen,
- Korrelationsanalysen.

Alle diese Analysen geben wichtige Einblicke in die Daten und sind unverzichtbar, insbesondere auch im Hinblick auf weiterführende Analysen wie der Regelgenerierung mit integrierter Klassifikation.

**Grenzwertverletzungen:** Da die Daten von einer größeren Zahl von Dokumentaren eingegeben wurden, entstanden in natürlicher Weise Fehleingaben wie Tippfehler, deren Korrektur sich im Nachhinein nur teilweise bewerkstelligen lässt. Deshalb wurden alle nicht korrigierbaren Werte, die Grenzwerte verletzten, bei den Analysen weggelassen. In der Regel waren das pro Variable weniger als 1% der Werte. Bei zwei Variablen Albumin (14,1% Grenzwertverletzungen) und CalciumSerum (57,6% Grenzwertverletzungen) war das Entfernen einzelner fehlerhafter Werte nicht möglich: Beim CalciumSerum wurde während der Dokumentation die Existenz zweier verschiedener Messverfahren nicht berücksichtigt. Die hohe Anzahl von fehlerhaften Albuminwerten ist nicht erklärbar.

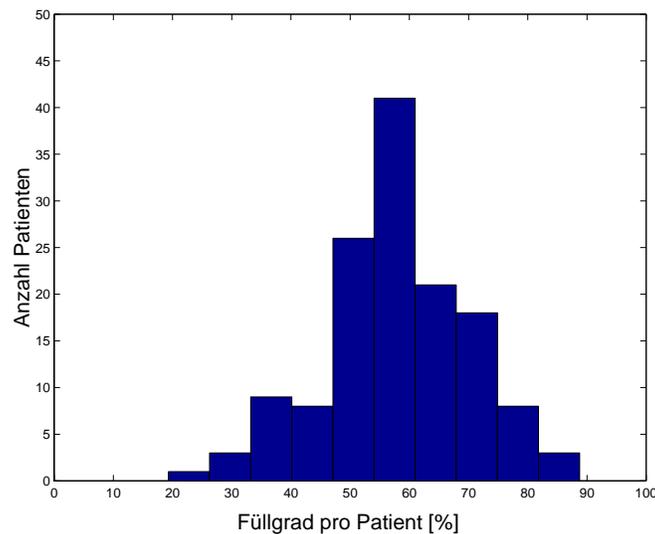
**Füllgrad der Daten (in Abhängigkeit des Samplings):** Da alle Daten von Ärzten zu beliebigen Uhrzeiten dokumentiert werden, ist ein Sampling der Daten notwendig. Treten mehrere Daten innerhalb eines Sampling-Intervalls auf, so wurde der Mittelwert dieser Daten gewählt. Probiert wurden Sampling-Intervalle von 24h, 12h, 8h, 6h, 4h, 2h und 1h. Die meisten Variablen haben bei einem Sampling  $\leq 12$ h einen zu niedrigen Füllgrad (also zu viele fehlende Werte), um sinnvolle Analysen durchführen zu können. Ausnahmen sind der BlutdruckSystolisch, die Herzfrequenz, die Temperatur und die Urinmenge, die zumindestens tagsüber oft stündlich dokumentiert wurden. Die Tabelle 7.1 zeigt den Füllgrad der 10 am häufigsten und der 5 am seltensten gemessenen Variablen (ohne Medikamente/Therapien). Bei Medikamentengaben würde der Füllgrad nur die Häufigkeit der Verabreichung wiedergeben, nicht die Messqualität. Zusätzlich sieht man in der Tabelle die Anzahl der Patienten (maximal 138), bei denen überhaupt Messungen vorlagen. Für die Analysen wird als Standard von einem 24h-Sampling ausgegangen, wenn nichts anderes gesagt wird.

Selbst die am häufigsten gemessenen Variablen haben bei einem 24h-Sampling (datentechnisch) eine recht hohe Anzahl von fehlenden Werten, so dass man nicht von einzelnen fehlenden Werten sprechen kann. Eine Verbesserung dieser Problematik kann in Zukunft nur durch prospektiv geplante Messungen verbessert werden.

Variablen, für die keinerlei Werte vorhanden waren, sind: Leukozytenkonzentrat, Organtransplantation und DobutaminEinmalgabe. Diese Variablen wurden nicht bei der Analyse berücksichtigt.

Variable	Häufigkeit in %	Anzahl Patienten
systolischer Blutdruck	96.85	138
Herzfrequenz	96.70	138
diastolischer Blutdruck	94.53	134
Temperatur	93.49	137
Hämoglobin	90.64	137
Hämatokrit	90.44	138
Leukozyten	88.52	136
KaliumSerum	87.78	137
Urinmenge	87.33	131
NatriumSerum	87.14	137
Thrombozyten	85.51	135
...	...	...
Cholesterin	9.81	54
Harnsäure	6.65	43
PCWP	6.41	29
Eisen	2.66	14
PPSB	0.44	9

**Tabelle 7.1:** Klinikdaten: Messhäufigkeit der 10 häufigsten und der 5 seltensten Messungen nach einem 24h-Sampling für die septischen Schock-Patienten der Klinikdaten. Wurde der systolische Blutdruck gemessen, so liegt i. Allg. auch eine Messung des diastolischen und damit ein Wert des errechneten mittleren Blutdrucks vor. Die Anzahl der Patienten, für die Messungen vorlagen, ist in der 2. Spalte angegeben.



**Abbildung 7.1:** Klinikdaten: Füllgrad der Daten pro Patient.

**Füllgrad der Patienten:** Pro Patient kann man (nach einem 24h-Sampling) die Anzahl der vorhandenen Messdaten der Anzahl der maximal möglichen Messungen innerhalb der Liegezeit gegenüberstellen. Dieser Füllgrad der Messungen pro Patient ist als Histogramm in der Abb. 7.1 dargestellt.

**Zeitreihendynamik:** Die Häufigkeit der Messungen pro Variable über die Zeit betrachtet kann als verbessert gegenüber den Vorstudierendaten angesehen werden. Die einzige Variable, die keine adäquate Messhäufigkeit über die Zeit hat, ist die Harnsäure (nur etwa alle 5 Tage gemessen).

**Ausschluss von Variablen, Verwendbarkeit der Variablen:** Grundsätzlich von der Analyse ausgeschlossen werden: CalciumSerum und Albumin (zu hoher Anteil an Grenzwertverletzungen), Leukozytenkonzentrat, Organtransplantation, Dobutamin-Einmalgabe (nicht vorgekommen), Harnsäure (keine adäquate Zeitreihendynamik), THAMTris, GCSF, HCLLösung, DopaminEinmalgabe (bei weniger als 5 Patienten verabreicht), Eisen (bei weniger als 28 Patienten gemessen), PCWP, Cholesterin (schlechte Samplingeigenschaften, d.h. mal sehr häufig, dann wieder gar nicht gemessen; insgesamt bei wenigen Patienten gemessen).

Von den nicht ausgeschlossenen Daten können die Messwerte (Datentyp „WERT“) prinzipiell einer metrischen Regelgenerierung zugeführt werden. Das gleiche gilt für die Dosen der Medikamente (Datentyp „DOSIS“) Schleifendiuretika, Pentaglobin, Kristalloide, Kolloide, EK, FFP, AdrenalinPerfusor, NoradrenalinPerfusor, DopaminPerfusor, DobutaminPerfusor und O<sub>2</sub>Gabe. Zusätzlich werden die Dosen als symbolische binäre Variablen (pro Patient ohne Zeitabhängigkeit) kodiert, um sie einer symbolischen Auswertung zukommen zu lassen. Die binären symbolischen Variablen werden pro Patient ausgewertet. Bei den Variablen Lungeninfiltration, Analgetika, Antiarrhythmika, Beatmungsform und Katheter werden zusätzlich verschiedene Unterklassen kodiert.

**Verteilungseigenschaften der metrischen Variablen, Behandlung fehlender Werte:** Die meisten Messvariablen mit stetigen Verteilungen besitzen annähernd Normalverteilungen. Ausnahmen sind: O<sub>2</sub>-Sättigung (links-schief bzw. rechtssteil), Thrombozyten, TZ, GPT, GGT, AlkalischePhosphatase, BilirubinGesamt, CRP, Urinmenge (rechts-schief bzw. linkssteil). Da für die Ausnahmen die Schätzung fehlender Werte auf der Basis von verrauschten Mittelwerten nicht geeignet ist, wird die im Abschnitt 3.3 diskutierte Strategie „mediannarrownoise“ verwendet.

**Epidemiologische Auszählungen (analog zu [WBH98]):** Die Auszählungen werden in der Form (alle Patienten/überlebende Pat./verstorbene Pat.) als Mittelwerte – wenn nicht anders gesagt – angegeben. Geschlecht: männlich (55.8 / 48.5 / 62.9) [%], weiblich (44.2 / 51.5 / 37.1) [%]; Alter: (66.7 / 66.0 / 67.4) [Jahre]; Gewicht männlich (79.0 / 82.0 / 76.7) [kg]; Gewicht weiblich (66.8 / 67.6 / 65.7)

[kg]; Größe männlich (1.74 / 1.75 / 1.74) [m]; Größe weiblich (1.62 / 1.63 / 1.61) [m]; Liegedauer (14.7 / 15.0 / 14.4) [Tage]; Minimum Liegedauer (2 / 4 / 2) [Tage]; Maximum Liegedauer (45 / 41 / 45) [Tage], Beatmungstage der beatmeten Patienten (9.9 / 10.3 / 9.6) [Tage], Tage der Hämofiltration bei Hämofiltrations-Patienten (7.1 / 7.4 / 2.0) [Tage] (es gab nur einen überlebenden Patienten, bei dem eine Hämofiltration angewendet wurde). Von 11 Patienten mit ARDS sind 10 verstorben (90.91%). Aufgrund der Fallzahl ist der prozentuale Anteil aber ungenau; würde ein 12. ARDS-Patient nicht versterben, so entsprächen dann 10 von 12 Verstorbene nur noch 83.33%. Teilt man die 138 Patienten in 4 Gruppen gemäß der Dokumentationsreihenfolge, so nahm die mittlere Liegedauer von 20.0 Tagen im ersten Viertel auf 12.1 im vierten Viertel ab, da aufgrund der Zeitknappheit bei der Dokumentation am Ende eher Patienten mit weniger Liegetagen dokumentiert wurden.

**Diagnosen:** Bei den Diagnosen ergab sich das etwas unbefriedigende Bild, dass nur wenige Diagnosen häufig waren. Z.B. trat eine Immunsuppression nur bei 5 Patienten auf, die Untergruppen Bestrahlung, Chemotherapie gar nicht. Um Diagnosen mit in die symbolische Regelgenerierung zu integrieren, wurde eine (bereits geringe) Mindestanzahl von 11 Patienten (7.97%) gefordert, da ja noch eine Aufteilung in Trainings- und Testdaten berücksichtigt werden muss. Diagnosen, die für mind. 11 Patienten zutrafen, waren: akute Herzerkrankung (Untergruppe Rhythmusstörung, 16 Pat.), chronische Herzerkrankung (Untergruppe Herzinsuffizienz sowie KHK und Hypertonus, 18, 17 bzw. 36 Pat.), akutes Nierenversagen (26 Pat.), chronische Nierenerkrankung (Untergruppe kompensierte Niereninsuffizienz, 11 Pat.), akute Lungenerkrankung (Untergruppe ARDS, 11 Pat.), chronische Lungenerkrankung (Untergruppe COPD, 13 Pat.), chronische Lebererkrankung (11 Pat.), keine der 30 Diagnosen (35 Pat.).

**Operationen:** Die geringen Fallzahlen sind bei den Operationen noch problematischer als bei den Diagnosen. Die meisten Operationen sind im hierarchisch organisierten vierstelligen OP301-Code Einzelfälle. Selbst eine Gruppierung ohne Berücksichtigung der vierten Ziffer erzeugt keine großen Fallzahlen. Folgende OP301-Codes, die bei mehr als 10 Patienten auftraten, wurden für die symbolische Regelgenerierung berücksichtigt: 5311 (15 Pat.), 543 (vierte Stelle im OP301-Code beliebig, 21 Pat.), 5454 (20 Pat.), 5455 (40 Pat.), 5467 (12 Pat.), 5469 (21 Pat.), 5511 (18 Pat.), 5541 (48 Pat.), 5545 (11 Pat.), 8176 (22 Pat.). Mit Ausnahme der „543“ wurden nur vierstellige OP301-Codes berücksichtigt, da dreistellige Codes i. Allg. nicht spezifisch genug für eine Regelgenerierung sind.

Die Hoffnung *Kombinationen* aus Diagnosen und Operationen auszuwerten, wird sich bei der symbolischen Regelgenerierung aufgrund der geringen Fallzahl als nicht möglich herausstellen. Um z.B. 50 ARDS-Patienten zur genaueren Bestimmung der o.g. prozentualen Anteile zu bekommen, würde man über 600 Patienten benötigen (denn  $50/11 \cdot 138 > 600$ ).

**Korrelationsanalysen auf den Rohdaten:** Eine Korrelationsanalyse zwischen den Messvariablen bzw. Medikamenten und verstorben/überlebt (Exitus-Variable) ergab bei 7 Variablen einen Wert, der betragsmäßig größer als 0.3 war, bei einem Mindestfüllgrad von 20% (nach 24h-Sampling) für die Variablen. Alle Werte blieben aber absolut unter 0.4, so dass keine eindeutigen Beziehungen zwischen den Variablensamples und der Exitus-Variable nachgewiesen werden konnten. Die 7 Variablen waren: Thrombozyten (-0.31), TPT (-0.39), DopaminPerfusor (0.31), Peak (0.32), FiO<sub>2</sub> (0.31), SOFA (0.34) und MODS (0.36).

Korrelationswerte zwischen zwei Variablen, die absolut größer als 0.9 waren und damit einen starken Zusammenhang kennzeichnen (mind. 20% Füllgrad pro Variable nach 24h-Sampling, mind. 25 Messpaare): BaseExcess/Bicarbonat (0.93), Erythrozyten/Hämatokrit (0.91), SOFA/MODS (0.91). Bei weiteren 27 Paaren wurden Korrelationen absolut  $> 0.5$  festgestellt. Die hohe Korrelation zwischen BaseExcess und Bicarbonat erklärt, dass in der Tab. 7.2 ähnliche Korrelationswerte für BaseExcess und Bicarbonat mit GOT und GPT auftreten.

Die interessantesten Korrelationsergebnisse erhält man, wenn man Korrelationen zwischen Variablen, getrennt für Überlebende und Verstorbene berechnet und die Differenzen betrachtet. Daraus könnte man (bei sehr häufigen Messungen, z.B. mind. stündlich) ein Alarmsystem bauen, das nur auf Korrelationsberechnungen beruht, siehe die Diskussion im Abschnitt 5.3. Ggf. könnte man bei häufigeren Messungen mit diesen Werten auch ein neuronales Netz trainieren. Analog zu den Tabellen 5.6 bis 5.8 werden die Ergebnisse dieser Korrelationsanalyse in den Tabellen 7.2 bis 7.4 wiedergegeben.<sup>1</sup> Es wurde aber eine absolut höhere Differenz und ein höherer Füllgrad gefordert als bei den Vorstudien Daten.

## 7.2 Regelgenerierung mit Fuzzy-RecBF-DDA-Netzen

In diesem Abschnitt wende ich den modifizierten Fuzzy-RecBF-DDA-Regelgenerator auf die metrischen Klinikdaten an, um anhand der hier erhaltenen Klassifikations- und Regelergebnisse im Abschnitt 7.5 ein Alarmsystem aufbauen zu können. Zur Anwendung kommen die Inhalte des Kapitels 3.

Eine wichtige Frage, die dabei beantwortet wird, ist die Frage nach den geeigneten Variablen für ein Alarmsystem. Wie kommt man von den 103 metrischen Variablen zu sinnvollen Ergebnissen? Klar ist, dass aufgrund der inhomogenen und der teilweise schlechten Datenqualität einzelner Variablen (insbesondere durch seltene

---

<sup>1</sup>Die Signifikanzniveaus wurden mit SPSS 9.0 (bivariate zweiseitige Pearsonsche Korrelation) berechnet.

Variable X	Variable Z	$\widetilde{\text{COR}}(X_s, Z_s)$	$\widetilde{\text{COR}}(X_d, Z_d)$	Differenz
Bicarbonat	GPT	-0.52	0.27	0.79
pCO <sub>2</sub> Arteriell	Erythrozyten	-0.12	0.57	0.69
Bicarbonat	GOT	-0.34	0.27	0.61
BaseExcess	GPT	-0.40	0.15	0.55
pH	Harnstoff	-0.37	0.17	0.54
BaseExcess	GOT	-0.34	0.20	0.54

**Tabelle 7.2:** Klinikdaten: Korrelationen zwischen zwei Messvariablen mit den höchsten Korrelationsdifferenzen  $> 0.5$  zwischen Verstorbenen und Überlebenden (alle Tage, alle Patienten). Differenz bedeutet  $|\widetilde{\text{COR}}(X_s, Z_s) - \widetilde{\text{COR}}(X_d, Z_d)|$ . Die Messhäufigkeiten der Variablen sind in der Tabelle 7.3 angegeben. Es wurde mind. 40% Messhäufigkeit gefordert.

Variable	Häufigkeit in %
Erythrozyten	73.3
Harnstoff	72.4
pH	71.7
pCO <sub>2</sub> Arteriell	68.8
BaseExcess	65.9
Bicarbonat	65.5
GOT	43.7
GPT	42.1

**Tabelle 7.3:** Klinikdaten: Messhäufigkeit der Variablen aus der Tab. 7.2 (alle Patienten der Klinikdaten, alle Tage, kein Sampling).

Variable X	Variable Z	$\#(X_s, Z_s)$	$\#(X_d, Z_d)$	$s_s$	$s_d$
Bicarbonat	GPT	57	75	<b>0.01</b>	<b>0.05</b>
pCO <sub>2</sub> Arteriell	Erythrozyten	152	217	-	0.01
Bicarbonat	GOT	66	81	<b>0.01</b>	<b>0.05</b>
BaseExcess	GPT	63	75	0.01	-
pH	Harnstoff	181	204	<b>0.01</b>	<b>0.05</b>
BaseExcess	GOT	76	83	0.01	-

**Tabelle 7.4:** Klinikdaten: Anzahl der Datensamples und Signifikanzniveau der Korrelationen der Tab. 7.2  $s_s := \text{Signifikanzniveau}(\widetilde{\text{COR}}(X_s, Z_s))$  und  $s_d := \text{Signifikanzniveau}(\widetilde{\text{COR}}(X_d, Z_d))$ . Gilt  $s_s \leq 0.05$  und  $s_d \leq 0.05$ , so sind die Einträge fett.



**Häufig16K:**<sup>2</sup> Der Datensatz besteht aus den 16 am häufigsten gemessenen Variablen der Klinikdaten (Füllgrad pro Variable mind. 80%), d.h. Herzfrequenz [1/min], systolischer Blutdruck [mmHg], diastolischer Blutdruck [mmHg], Temperatur [°C], zentralvenöser Druck (ZVD) [mmHg], O<sub>2</sub>-Sättigung [%], Leukozyten [1000/ $\mu$ l], Hämoglobin [g/dl], Hämatokrit [%], Thrombozyten [1000/ $\mu$ l], PTT [s], NatriumSerum [mmol/l], KaliumSerum [mmol/l], Kreatinin [mg/dl], Blutzucker [mg/dl], Urinmenge [ml].

Die Anzahl der Samples, die pro Datensatz in Abhängigkeit des betrachteten Zeitraums zur Verfügung standen, sind für alle Datensätze in der Tab. 7.5 wiedergegeben. Dabei wird auch die Anzahl der Variablen angegeben, die mindestens pro Sample gemessen sein mussten.

Es folgen Datensätze, die aus den medizinischen Subsystemen gebildet wurden.

**Herz10K:** Der Datensatz besteht aus den 10 gemessenen Variablen des Herzsystems der Klinikdaten, d.h. Herzfrequenz [1/min], systolischer Blutdruck [mmHg], diastolischer Blutdruck [mmHg], zentralvenöser Druck (ZVD) [mmHg], Kristalloide [ml], Kolloide [ml], AdrenalinPerfusor [ $\mu$ g/kg/min], NoradrenalinPerfusor [ $\mu$ g/kg/min], DopaminPerfusor [ $\mu$ g/kg/min], DobutaminPerfusor [ $\mu$ g/kg/min].

**Blut14K:** Der Datensatz besteht aus den 14 gemessenen Variablen des Blutsystems der Klinikdaten, d.h. Leukozyten [1000/ $\mu$ l], Erythrozyten [1000/ $\mu$ l], Hämoglobin [g/dl], Hämatokrit [%], Thrombozyten [1000/ $\mu$ l], TPZ [%], PTT [s], TZ [s], AT3 [%], Fibrinogen [mg/dl], Gesamtprotein [g/dl], Blutzucker [mg/dl], EK [ml], FFP [ml].

**Lunge11K:** Der Datensatz besteht aus den 11 gemessenen Variablen des Lungensystems der Klinikdaten, d.h. pO<sub>2</sub>Arteriell [mmHg], pCO<sub>2</sub>Arteriell [mmHg], BaseExcess [-], Bicarbonat [mmol], O<sub>2</sub>Sättigung [%], O<sub>2</sub>Gabe [l/min], Peak [cmH<sub>2</sub>O], I:E [-], Beatmungsfrequenz [1/min], FiO<sub>2</sub> [%], PEEP [mmHg].

**Infekt7K:** Der Datensatz besteht aus den 7 gemessenen Variablen des Infektionssystems der Klinikdaten, d.h. Temperatur [°C], Leukozyten [1000/ $\mu$ l], Thrombozyten [1000/ $\mu$ l], AT3 [%], Fibrinogen [mg/dl], CRP [mg/l], Pentaglobin [ml].

**Leber13K:** Der Datensatz besteht aus den 13 gemessenen Variablen des Lebersystems der Klinikdaten, d.h. TPZ [%], PTT [s], TZ [s], AT3 [%], GOT(ASAT) [U/l], GPT(ALAT) [U/l], GGT [U/l], AlkalischePhosphatase [U/l], BilirubinGesamt [mg/dl], LDH [U/l], Triglyceride [g/dl], Gesamtprotein [g/dl], Hämofiltration [ml].

---

<sup>2</sup>Das „K“ steht im Folgenden für *Klinikdaten*; die Zahl bezeichnet die Anzahl der Variablen.

Datensatz	Zeitraum	$ S $ insges.	verwendete $ S $	Anteil [%]	$ V $
Häufig16K	ALLE	2029	1710	84.3	13
Häufig16K	E3	413	355	86.0	13
Häufig16K	L5	675	533	79.0	13
Häufig16K	L3	413	303	73.4	13
Häufig16K	L2	276	181	65.6	13
Häufig16K	L1	138	66	47.8	13
Häufig16K	E8S6	413	365	88.4	13
Häufig16K	E4S2	413	359	86.9	13
Häufig16K	S3	411	394	95.9	13
Herz10K	ALLE	2029	1951	96.2	9
Herz10K	L3	413	402	97.3	9
Blut14K	L3	413	244	59.1	10
Lunge11K	L3	413	255	61.7	8
Infekt7K	L3	413	260	63.0	5
Leber13K	L3	413	123	29.8	6
Niere9K	L3	413	282	68.3	6
SB6K	L3	413	237	57.4	5
Wichtig4K	L3	413	407	98.5	4
Wichtig6K	L3	413	323	78.2	5
Wichtig8K	L3	413	351	85.0	6
BeatKat3K	L3	413	413	100.0	3
SOFA1K	L3	413	273	66.1	1
SOFA1K	S3	411	402	97.8	1
APACHE1K	L3	413	273	66.1	1
SAPS1K	L3	413	273	66.1	1
MODS1K	L3	413	273	66.1	1
SystDiasThrombo3K	L3	413	400	96.9	2
DiasSys2K	L3	413	391	94.7	2
DiasThrombo2K	L3	413	304	73.6	2
SystThrombo2K	L3	413	313	75.8	2
SystThrombo2K	S3	411	378	92.0	2
RRDiastolisch1K	L3	413	391	94.7	1
RRSystolisch1K	L3	413	405	98.1	1
ZVD1K	L3	413	302	73.1	1
Thrombozyten1K	L3	413	316	76.5	1

**Tabelle 7.5:** Klinikdaten: Anzahl und prozentualer Anteil der verwendeten Samples.  $|S|$  = Anzahl der Samples. Anzahl  $|V|$  der Variablen, die mindestens pro Sample gemessen sein mussten. Die Notation des Zeitraums ist im Anschluss an die Beschreibung der Datensätze auf S. 207 in diesem Abschnitt erläutert.

**Niere9K:** Der Datensatz besteht aus den 9 gemessenen Variablen des Nierensystems der Klinikdaten, d.h. NatriumSerum [mmol/l], KaliumSerum [mmol/l], ChloridSerum [mmol/l], Kreatinin [mg/dl], Harnstoff [mg/dl], Urinmenge [ml], Schleifendiuretika [mg], Hämofiltration [ml], ParenteraleErnaehrung [ml].

**SB6K:** Der Datensatz besteht aus den 6 gemessenen Variablen des Säure-Base-Systems der Klinikdaten, d.h. pH [-], pO<sub>2</sub>Arteriell [mmHg], pCO<sub>2</sub>Arteriell [mmHg], BaseExcess [-], Bicarbonat [mmol], KaliumSerum [mmol/l].

Die nächsten drei Datensätze wurden aus wichtigen Variablen des Datensatzes Häufig16K zusammengestellt, vgl. Tab. 7.10; Wichtig8K enthält zusätzlich die Variablen des Datensatzes BeatKat3K.

**Wichtig4K:** Der Datensatz besteht aus 4 gemessenen Variablen der Klinikdaten, die eine hohe Wichtigkeit haben, d.h. Herzfrequenz [1/min], systolischer Blutdruck [mmHg], zentralvenöser Druck (ZVD) [mmHg], Thrombozyten [1000/ $\mu$ l].

**Wichtig6K:** Der Datensatz besteht aus 6 gemessenen Variablen der Klinikdaten, die eine hohe Wichtigkeit haben (siehe Tab. 7.10), d.h. Herzfrequenz [1/min], systolischer Blutdruck [mmHg], diastolischer Blutdruck [mmHg], zentralvenöser Druck (ZVD) [mmHg], Thrombozyten [1000/ $\mu$ l], NatriumSerum [mmol/l].

**Wichtig8K:** Der Datensatz besteht aus 8 gemessenen Variablen der Klinikdaten, die eine hohe Wichtigkeit haben, d.h. systolischer Blutdruck [mmHg], diastolischer Blutdruck [mmHg], zentralvenöser Druck (ZVD) [mmHg], Leukozyten [1000/ $\mu$ l], Thrombozyten [1000/ $\mu$ l], DopaminPerfusor [ $\mu$ g/kg/min], DobutaminPerfusor [ $\mu$ g/kg/min], FiO<sub>2</sub> [%].

**BeatKat3K:** Der Datensatz besteht aus 3 gemessenen Variablen der Klinikdaten, die der Beatmung bzw. den Katecholaminen zugerechnet werden können, d.h. DopaminPerfusor [ $\mu$ g/kg/min], DobutaminPerfusor [ $\mu$ g/kg/min], FiO<sub>2</sub> [%]. Die Beatmungs- und Medikamentvariablen haben generell keine fehlenden Werte, da bei Nichtgabe oder Nichtbeatmung die Variable auf 0 gesetzt wird. Diese drei Variablen wurden im Lungen- bzw. Herzsystem als besonders wichtiger Beatmungsparameter bzw. besonders wichtige Katecholamine eingestuft (durch das Maß der „Wichtigkeit“).

Es folgen die Datensätze der Score-Variablen.

**SOFA1K:** Der Datensatz besteht aus der in der MEDAN-Datenbank berechneten SOFA-Variablen der Klinikdaten. Der SOFA-Score ist aus 6 Einzelscores zusammengesetzt, die wiederum aus insgesamt 10 Variablen berechnet werden; der GCS (Glasgow Coma Score) ging nicht in die Berechnung mit ein.

Bei diesem Datensatz wurden die Einzelvariablen mit einer „medizinischen“ fehlenden Werte-Strategie behandelt. Fehlende Variablen wurden innerhalb eines medizinischen „Gültigkeitszeitraums“ (in diesem Zeitraum verändert sich der Wert der Variable nur selten stark) mit dem letzten gemessenen Wert ersetzt. Waren dann noch fehlende Werte vorhanden, so wurde er als normal angenommen und so bekam dieser Einzelscore 0 Punkte. Für die vorhandenen Werte, wurde der Wert pro Tag ausgewählt, der die meisten Punkte bekommt („WorstSOFA“). Diese Strategie ist medizinisch gerechtfertigt. Der absolute Wert des „WorstSOFA“ ist vom Sampling abhängig [SKRT00]. Die SOFA-Variable wurde als vorberechnete Einzelvariable aus der MEDAN-Datenbank übernommen.

**APACHE1K, SAPS1K, MODS1K:** Der jeweilige Datensatz besteht aus der in der MEDAN-Datenbank berechneten APACHE II-, SAPS II- bzw. MODS-Variablen der Klinikdaten. Der APACHE II-, der SAPS II- und der MODS-Score sind im Abschnitt 2.1.3 beschrieben. Für die Berechnung des APACHE II-, SAPS II- bzw. MODS-Scores gelten bez. der fehlenden Werte die Bemerkungen zum SOFA1K-Datensatz.

Es folgen einige Datensätze, die insbesondere die Rolle der Blutdrücke und der Thrombozyten beleuchten helfen sollen.

**SystDiasThrombo3K:** Der Datensatz besteht aus 3 gemessenen Variablen der Klinikdaten: diastolischer Blutdruck [mmHg], systolischer Blutdruck [mmHg] und Thrombozyten [1000/ $\mu$ l].

**DiasSys2K:** Der Datensatz besteht aus 2 gemessenen Variablen der Klinikdaten: diastolischer Blutdruck [mmHg] und systolischer Blutdruck [mmHg].

**DiasThrombo2K:** Der Datensatz besteht aus 2 gemessenen Variablen der Klinikdaten: diastolischer Blutdruck [mmHg] und Thrombozyten [1000/ $\mu$ l].

**SystThrombo2K:** Der Datensatz besteht aus 2 gemessenen Variablen der Klinikdaten: systolischer Blutdruck [mmHg] und Thrombozyten [1000/ $\mu$ l].

**RRDiastolisch1K:** Der Datensatz besteht aus der gemessenen Variable diastolischer Blutdruck [mmHg] der Klinikdaten.

**RRSystolisch1K:** Der Datensatz besteht aus der gemessenen Variable systolischer Blutdruck [mmHg] der Klinikdaten.

**ZVD1K:** Der Datensatz besteht aus der gemessenen Variable ZVD [mmHg] der Klinikdaten.

**Thrombozyten1K:** Der Datensatz besteht aus der gemessenen Variable Thrombozyten [1000/ $\mu$ l] der Klinikdaten.

Die *Zeitabschnitte* der Patientenzitreihen, die verwendet wurden, seien mit folgenden Kürzeln bezeichnet: alle Tage (ALLE), die letzten  $n$  Tage ( $Ln$ ), die ersten 3 Tage (E3), 3 Tage ab dem 1. Tag des Auftretens des septischen Schocks (S3), ausgehend vom letzten Tag den  $(-n)$ -ten bis  $(-n - 2)$ -ten Tag des Intensivaufenthalts ( $En + 2Sn$ ),  $n \in \mathbb{N} \cup \{0\}$ .

Es hat sich herausgestellt, dass am besten mit Datensätzen trainiert werden kann, die Samples von mindestens 3 (aufeinanderfolgenden) Tagen enthalten. Weniger als 3 Tage können zu einem zu hohen Datenverlust durch fehlende Werte führen und damit verbunden zu möglichen Verzerrungen hin zu scheinbar besseren bzw. schlechteren Ergebnissen. Diese Situation könnte sich durch die Verwendung von deutlich mehr als 138 Patienten verbessern. Da ein Patient nur 2 Liegetage hatte, wurden in diesem Fall nur seine beiden Liegetage für die Analysen verwendet. Analysen von Daten mit mehr als 3 Tagen spiegeln nicht mehr gut die Zeitabhängigkeit wider, z.B. ALLE oder L5.

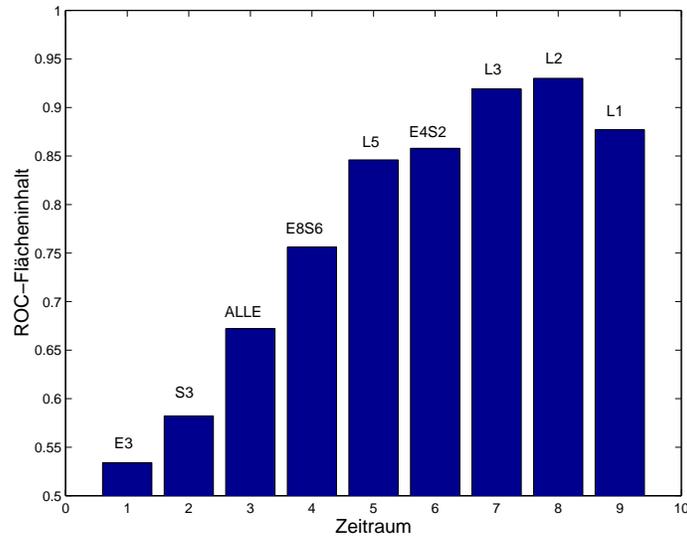
Alle Versuchsergebnisse sind Mittelwerte aus jeweils 5 Versuchen (jeweils zufällige, andere Wahl von Trainings- und Testdaten). In der Testmenge befanden sich keine Daten von Patienten, die Daten in der Trainingsmenge hatten (disjunkte Patientengruppen).

**Zusammenfassende Diskussion der Ergebnisse:** In der Tab. 7.6 ist die Klassifikationsleistung der hier dokumentierten Versuche im Überblick zusammengestellt. Die Abb. 7.2 illustriert die Ergebnisse der Häufig16K-Daten in Abhängigkeit der Zeit. Die Abb. 7.3 vergleicht die Ergebnisse, die zum Zeitraum L3 gehören. Im Anschluss folgt eine Diskussion der Ergebnisse. Detailliertere Ergebnisse bez. der fünf interessanten Datensätze Häufig16K, SOFA1K, SystThrombo2K, Wichtig6K und SystDiasThrombo3K zum Zeitraum L3 werden nach der Ergebnisdiskussion wiedergegeben. Der nicht an Detailliertere interessierte Leser kann am Ende dieses Abschnitts zum Abschnitt über symbolische Regelgenerierung oder zum Abschnitt 7.5 über das Alarmsystem weiterblättern.

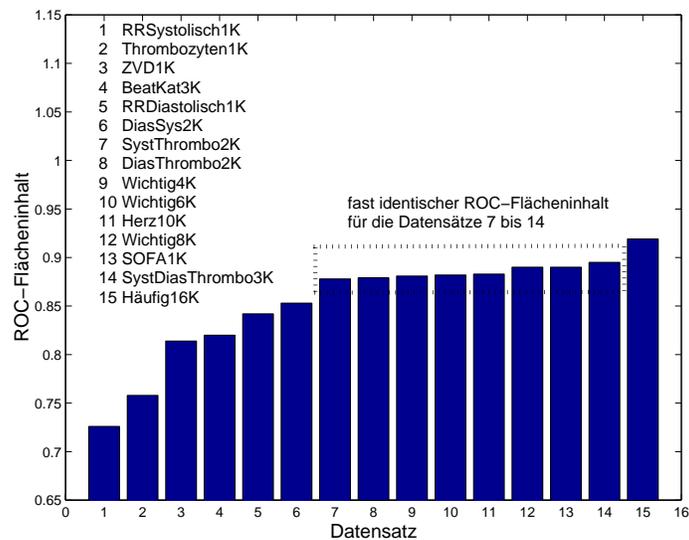
Die Analyse der Häufig16K-Datensätze zu verschiedenen Zeitpunkten (Abb. 7.2) zeigt deutlich die Verbesserung der Klassifikationsleistung zum Ende des Intensivaufenthalts hin. Vom Anfang zum Ende ist ein approximativ linearer Anstieg von einer fast zufälligen Klassifikation hin zu einer sehr guten Klassifikation zu sehen. Diese Verbesserung ist prinzipiell bei allen Datensätzen zu sehen, so dass die besten Ergebnisse hier in den letzten drei Tagen zu erwarten sind. Bei den letzten zwei Tagen oder beim letzten Tag gehen zu viele Samples durch nicht vorhandene Messungen verloren, um hier noch von repräsentativen Ergebnissen sprechen zu können.

Datensatz	Zeitraum	Anzahl Regeln	ROC	Epochen
Häufig16K	ALLE	68.2	0.672	6.2
Häufig16K	E3	40.4	0.534	7.0
Häufig16K	L5	25.0	0.846	5.6
Häufig16K	L3	16.2	0.919	3.8
Häufig16K	L2	12.4	0.930	3.4
Häufig16K	L1	5.0	0.877	3.8
Häufig16K	E8S6	23.8	0.756	4.0
Häufig16K	E4S2	21.0	0.858	5.8
Häufig16K	S3	34.2	0.582	7.0
Herz10K	ALLE	53.4	0.668	5.6
Herz10K	L3	14.0	0.883	3.2
Blut14K	L3	16.8	0.901	4.8
Lunge11K	L3	11.4	0.860	5.2
Infekt7K	L3	19.8	0.828	7.0
Leber13K	L3	14.8	0.748	6.8
Niere9K	L3	26.8	0.716	7.0
SB6K	L3	18.8	0.709	7.0
Wichtig4K	L3	15.3	0.881	3.6
Wichtig6K	L3	16.0	0.882	3.6
Wichtig8K	L3	11.4	0.890	3.4
BeatKat3K	L3	12.2	0.820	6.2
SOFA1K	L3	4.8	0.890	3.8
SOFA1K	S3	6.6	0.570	6.2
APACHE1K	L3	5.4	0.804	7.0
SAPS1K	L3	5.6	0.794	5.8
MODS1K	L3	4.4	0.787	4.2
SystDiasThrombo3K	L3	15.8	0.895	3.8
DiasSys2K	L3	16.0	0.853	4.8
DiasThrombo2K	L3	12.8	0.879	4.2
SystThrombo2K	L3	13.0	0.878	4.6
SystThrombo2K	S3	19.8	0.559	7.0
RRDiastolisch1K	L3	7.4	0.842	4.8
RRSystolisch1K	L3	5.6	0.726	7.0
ZVD1K	L3	8.6	0.814	6.2
Thrombozyten1K	L3	7.8	0.758	7.0

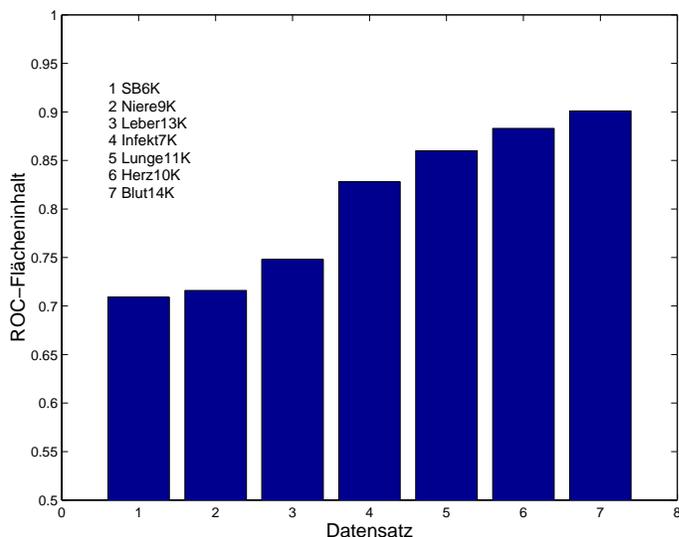
**Tabelle 7.6:** Klinikdaten: Klassifikationsleistung, Anzahl der Regeln und benötigte Epochenzahl als Mittelwert von 5 Versuchen auf verschiedenen Datensätzen. Die Epochenzahl wurde immer zwischen 3 und 7 begrenzt. Der ROC-Flächeninhalt ist mit drei Nachkommastellen angegeben; die Genauigkeit ist sicher nicht höher als  $\pm 0.02$ .



**Abbildung 7.2:** Häufig16K-Daten: ROC-Flächeninhalt in der Abhängigkeit vom Zeitraum der Klassifikationsversuche aller 5 Versuchswiederholungen. Da für die Zeiträume L2 und L1 (Stäbe 8 und 9) weniger Samples zur Verfügung standen, sind diese Ergebnisse weniger vertrauenswürdig.



**Abbildung 7.3:** Klinikdaten: ROC-Flächeninhalte der Versuche zum Zeitraum L3. Die Datensätze 7 bis 14 erzeugen einen fast identischen ROC-Flächeninhalt.



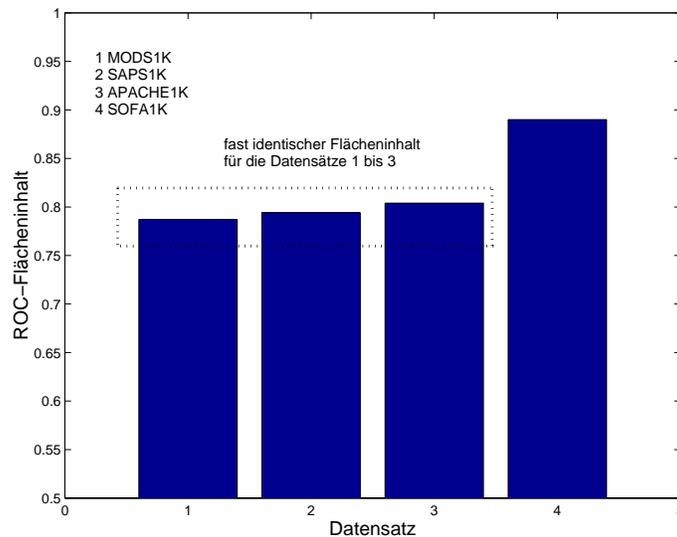
**Abbildung 7.4:** Klinikdaten: ROC-Flächeninhalte der Subsystem-Datensätze zum Zeitraum L3. Die Messhäufigkeiten waren allerdings recht unterschiedlich, siehe Tab. 7.5.

Leider ist es so, dass vermeintlich unwichtige Variablen seitens der Ärzte zum Ende des Intensivaufenthalts weniger dokumentiert werden als am Anfang. Interessant ist auch, dass an den letzten Tagen weniger Regeln erzeugt werden als an allen oder ersten Tagen (aufgrund der besseren Trennbarkeit der Daten).

In der Abb. 7.4 sind die Ergebnisse auf den Subsystemen gegenübergestellt. Oft enthalten die Subsysteme (mit Ausnahme von Herz10K) weniger häufig gemessene Variablen, so dass die Ergebnisse weniger vertrauenswürdig sind als von Herz10K oder von Datensätzen, die die Variablen aus verschiedenen Subsystemen zusammensetzen wie Häufig16K oder WichtignK. Zumindestens die Reihenfolge der Klassifikationsgüte erscheint medizinisch plausibel. Interessant ist aber durchaus, dass das Blutssystem etwas bessere Ergebnisse als das Herz- bzw. Lungensystem liefert.

Die Datensätze WichtignK,  $n = 4, 6, 8$  geben Versuche mit wichtigen Variablen aus dem Häufig16K-Datensatz wieder. Die Wichtigkeit der Variablen des Häufig16K-Datensatzes, berechnet nach der Methode aus dem Abschnitt 3.5, ist in der Tab. 7.10 im nächsten Abschnitt dokumentiert. Die Ergebnisse für  $n = 4, 6, 8$  sind fast identisch gut, siehe Abb. 7.3.

Der Datensatz BeatKat3K ( $\text{FiO}_2$  und 2 Katecholamine) liefert ein schlechteres Ergebnis (ROC-Fläche 0.820); ein weiterer (oben nicht genannter) Versuch mit 7 Variablen (d.h. mit allen 4 Katecholaminen Dopamin, Doputamin, Noradrenalin und Adrenalin sowie den 3 Beatmungsparametern Beatmungsfrequenz,  $\text{FiO}_2$  und Peak)

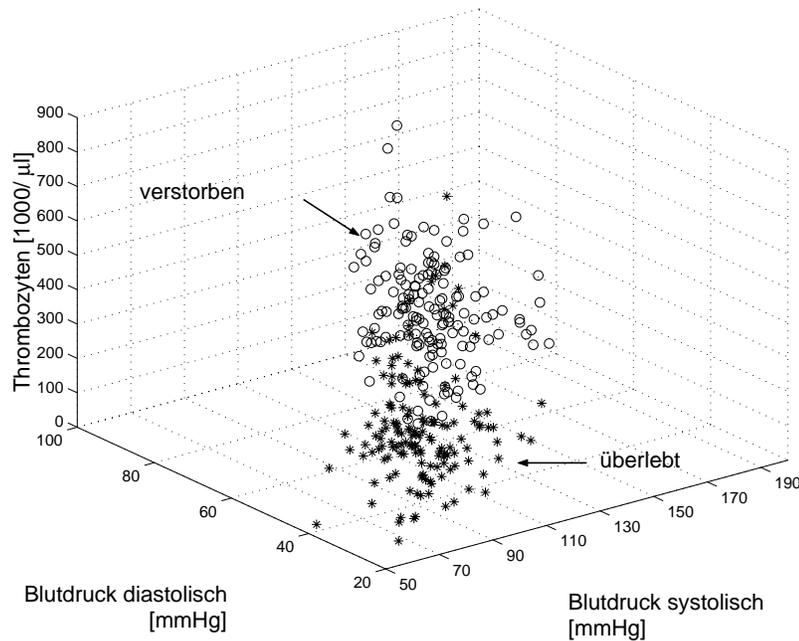


**Abbildung 7.5:** Klinikdaten: ROC-Flächeninhalte der Score-Datensätze zum Zeitraum L3. Der Datensatz SOFA1K erlaubt eine deutlich bessere Klassifikation.

lieferte auch ein schlechteres Ergebnis (ROC-Fläche 0.797). Siehe zu diesem Ergebnis auch die Diskussion weiter unten über Medikamente und Therapien im Vergleich zu körpereigenen Messwerten.

Vergleicht man die Ergebnisse der Score-Datensätze SOFA1K, APACHE1K, SAPS1K und MODS1K im Zeitraum L3 (Abb. 7.5), so stellt man fest, dass die drei Datensätze APACHE1K, SAPS1K und MODS1K eine ähnliche Klassifikationsleistung erlauben, während der SOFA1K-Datensatz zu einer deutlich besseren Klassifikation führt. Insofern ist es sinnvoll, die Ergebnisse der anderen Datensätze mit dem besten Score, dem SOFA-Score, zu vergleichen.

Die Ergebnisse, die mit dem SOFA-Score erzielt wurden sind erstaunlich gut, wenngleich etwas schlechter als die Ergebnisse des SystDiasThrombo3K- und des Häufig16K-Datensatzes. Nicht vergessen werden darf dabei, dass der SOFA-Score aus 10 Variablen berechnet wird, die nichtlinear in die Endsumme eingehen. Die Expertenmeinung, die dem SOFA zugrunde liegt ist nicht viel schlechter als das neuronale Netz. Auch berücksichtigt werden muss die eher optimistische fehlende Werte-Behandlung innerhalb des SOFA-Scores (siehe Beschreibung des Datensatzes SOFA1K weiter oben). Ein weiterer Aspekt ist, dass der SOFA-Score eine ähnliche, wenngleich einfachere, Berechnungsvorschrift verwendet wie ein neuronales Netz (1. Schicht: nichtlineare, unterschiedliche Stufenfunktionen für die Einzelvariablen mit Ausgang  $\in \{0, 1, 2, 3, 4\}$ , 2. Schicht: Summierung, Gewichte sind in diesem Fall konstant gleich 1).



**Abbildung 7.6:** SystDiasThrombo3K: Visualisierung aller Samples (Zeitraum L3), die keine fehlende Werte hatten. Die Kreise symbolisieren die Klasse „verstorben“, die Sterne die Klasse „überlebt“.

Interessant ist, dass die wichtigen Variablen SystolischerBlutdruck und Thrombozyten bereits eine identisch gute Klassifikation erlauben wie der SOFA-Score (2 statt 10 Variablen) und die 3 Variablen aus SystDiasThrombo3K sogar eine etwas bessere. In der Abb. 7.6 ist der dreidimensionale Datensatz SystDiasThrombo3K abgebildet. Man sieht die unterschiedliche Lage der Daten der verschiedenen Klassen; man sieht aber auch, dass aufgrund der Überlappungen keine Cluster zu erkennen sind.

Ergänzung: Die Ergebnisse in [H<sup>+</sup>00], wo der SOFA-Score u.a. für den ersten Tag des septischen Schocks berechnet wird, werden hier kurz als Vergleich diskutiert. Die Autoren führten eine ROC-Schätzung mittels SPSS 9.0 durch. Das Ergebnis lautete 0.70. Mein Ergebnis für diesen Zeitraum ist 0.570, ein schlechteres Ergebnis. Wie kann das sein? Die Autoren von [H<sup>+</sup>00] haben mir freundlicherweise zusätzliche Ergebnisse ihrer ROC-Analyse zukommen lassen, insbesondere das nicht publizierte 95%-Konfidenzintervall; es lautet [0.57, 0.83]. Weitere Analysen zeigten, dass Ergebnisse des neuronalen Netzes mit den unteren Grenzen der jeweiligen 95%-Konfidenzintervalle übereinstimmen. Die Erklärung hierfür ist, dass ich mit Trainings- und Testdaten (mit unbekanntem Testpatienten) sowie einer fehlenden Werte-Strategie arbeite und die ROC-Kurve auf Testdaten berechne, während SPSS 9.0 alle (vollständigen) Datensätze für eine ROC-Schätzung verwendet und somit die eigentliche Schätzung von 0.70 zu hoch liegt. Die Ergebnisse meiner Arbeit geben eher realistische Testbedingungen und Ergebnisse wieder, die für den Aufbau eines Alarmsystems für den realen Einsatz notwendig sind.

Sofort stellt sich die Frage, ob nicht sogar eine der drei Variablen SystolischerBlutdruck, DiastolischerBlutdruck oder Thrombozyten allein eine gute Klassifikation ermöglicht. Das ist aber nicht der Fall, siehe Abb. 7.3. Auch der weniger häufig gemessene ZVD liefert allein keine gute Klassifikation. Somit kann man die drei Variablen SystolischerBlutdruck, DiastolischerBlutdruck und Thrombozyten als wesentliche Merkmale *in Kombination* festhalten (als Merkmalskombination).

Daraus folgt zusammenfassend für das Alarmsystem im Abschnitt 7.5, dass sich ein Aufbau eines Alarmsystems und eine Überprüfung bzw. ein Vergleich des Alarmverhaltens vor allem für die 5 Datensätze Häufig16K, SOFA1K, SystDiasThrombo3K, SystThrombo2K,<sup>3</sup> Wichtig6K<sup>4</sup> (jeweils Zeitraum L3) lohnt (vgl. auch Abb. 7.3), für die im Folgenden Detailergebnisse präsentiert werden. Warum nimmt man nicht einfach den Datensatz mit den besten Klassifikationsergebnissen? Neben den eher kleinen Klassifikationsunterschieden, der bei diesen Datensätzen auftritt, ist auch die Anzahl der Regeln und der verwendeten Variablen sowie ein *stabiles, plausibles* Alarmverhalten über die Zeit interessant. Der Untersuchung des Alarmverhaltens widmet sich der Abschnitt 7.5. Die Hinzunahme weiterer Variablen zu den drei Variablen SystolischerBlutdruck und Thrombozyten verbessert zwar nicht unbedingt die Klassifikation, könnte aber aufgrund der zusätzlichen redundanten Informationen dazu beitragen, ein etwas stabileres Alarmverhalten zu erzeugen.

Grundsätzlich sinnvoll erweist es sich, nur Variablen zu verwenden, die körpereigene Messungen darstellen. Therapievariablen wie Medikamentendosen oder Einstellungen von Beatmungsparametern verbessern das Ergebnis nicht. Entscheidend für den Patienten ist z.B. nicht eine vom Arzt bestimmte Therapie an sich, sondern ob die Therapie wirkt, was sich besser durch körpereigene Variablen ausdrückt. Auf eine Untersuchung des Alarmverhaltens der Subsystem-Datensätze wurde verzichtet, da keine prinzipiell besseren Ergebnisse zu erwarten sind: das „beste“ Subsystem Blut14K liefert etwas schlechtere Ergebnisse als Häufig16K; die anderen Subsysteme liefern schlechtere Ergebnisse als SystDiasThrombo3K.

Welche Ergebnisse symbolische Variablen wie Diagnosen oder Operationen ermöglichen, wird im Abschnitt 7.3 untersucht, die Kombination mit den symbolischen Variablen im Abschnitt 7.4. Es folgen die Detailergebnisse, die diesen Abschnitt ergänzen.

---

<sup>3</sup>Man könnte auch DiastThrombo2K wählen; der systolische Blutdruck hatte aber eine geringfügig bessere Messhäufigkeit.

<sup>4</sup>An Stelle von Wichtig6K könnte auch Wichtig8K und/oder Wichtig4K betrachtet werden; das wird aber hier aus Platzgründen nicht ausgeführt.

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	88.68	2.27	86.58	92.05
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	84.02	4.44	78.91	90.91
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 7.7:** Häufig16K-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse ü	Klasse v
Klasse ü (Training)	45.98	5.95
Klasse v (Training)	5.37	42.70
Klasse ü (Test)	44.97	10.71
Klasse v (Test)	5.27	39.05

**Tabelle 7.8:** Häufig16K-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

## 7.2.2 Detaillierte Klassifikations- und Regelergebnisse

Alle Detaillierergebnisse beziehen sich auf den Zeitraum L3 (die letzten 3 Tage). Mit **Klasse ü** ist immer die Klasse der Daten von *überlebenden* Patienten, mit **Klasse v** die Klasse der Daten von *verstorbenen* Patienten gemeint. Generell werden für jeden Datensatz die Versuchseinstellungen und die Ergebnisse präsentiert und kommentiert.

### Detaillierergebnisse für den Häufig16K-Datensatz:

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 78$ , Ergebnisse siehe Tabellen 7.7 und 7.8 (Klassifikation) und Tabellen 7.9, C.21 und C.22 (Regelgenerierung). Die verwendeten Testdaten kamen in den 5 Testdurchläufen im Mittel zu 50.24% aus Klasse ü und zu 49.76% aus Klasse v. Die Epochenzahl betrug im Mittel 3.8 mit einer Standardabweichung von 0.84 (Minimum 3, Maximum 5).

Um das Klassifikationsergebnis im Hinblick auf die Veränderung der Klassifikationsschwelle besser einschätzen zu können, werden in der Abb. B.4 alle Klassifikationsergebnisse der 5 Versuche aufgetragen, wobei jeweils 10 Punkte eingetragen wurden, also insgesamt 50 Klassifikationen durchgeführt wurden. Berechnet man für die 5 ROC-Kurven die Flächeninhalte, so kann man den mittleren Flächeninhalt berechnen; er beträgt 0.5, falls die ROC-Kurve gerade die Diagonale darstellt und nähert sich dem Wert 1, falls die Kurve sich nach oben rechts „ausdehnt“. Für die Häufig16K-Daten erhält man den sehr guten Wert 0.919 (Min.: 0.893, Max.: 0.957).

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	8.2	1.6	6	10
Klasse v	8.0	1.4	7	10

**Tabelle 7.9:** Häufig16K-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

Zwei interessante Beispielregeln (Regeln 4 und 10) aus dem 1. der 5 Versuche (zum Niveau 0.0):

4) support class **survived** with test frequency **0.403** and test confidence **0.839** from **33** different patients if  
**var** *BlutdruckDiastolisch*  $\geq 59.52$  and  
**var** *Thrombozyten*  $\geq 27.00$  and  
**var** *KaliumSerum*  $\leq 4.70$

10) support class **deceased** with test frequency **0.292** and test confidence **0.933** from **24** different patients if  
**var** *BlutdruckSystolisch*  $\leq 139.14$  and  
**var** *ZVD*  $\geq 4.61$  and  
**var** *Haematokrit*  $\leq 34.70$  and  
**var** *Thrombozyten*  $\leq 271.50$  and  
**var** *KreatininSerum*  $\geq 0.71$

In der Abb. 7.7 ist die Konfidenz und in der Abb. 7.8 die Häufigkeit in Abhängigkeit vom Fuzzy-Niveau wiedergegeben. Auffallend ist, dass die mittlere Konfidenz erst ab dem Niveau 0.6 bzw. 0.9 für die Klasse ü bzw. Klasse v ansteigt, während vom Niveau 0.0 zu 0.1 ein starker Abfall zu erkennen ist. Für 16 Dimensionen scheinen die Daten von 138 Patienten immer noch nicht vollständig ausreichend zu sein, um eine bessere Konfidenzkurve zu erhalten.

In der Tab. 7.10 ist die Wichtigkeit der Variablen wiedergegeben. Man sieht, dass die Temperatur und die O<sub>2</sub>-Sättigung fast nicht relevant sind. Die Variablen Herzfrequenz, systolischer und diastolischer Blutdruck, ZVD, Thrombozyten und Natrium-Serum sind besonders wichtig; sie bilden die Grundlage der Datensätze WichtignK,  $n = 4, 6, 8$ . Der Datensatz Wichtig8K enthält allerdings zusätzlich die Variablen des Datensatzes BeatKat3K.

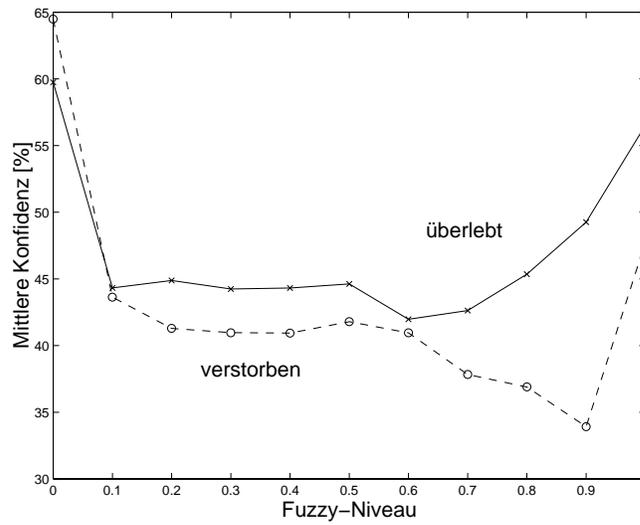


Abbildung 7.7: Häufig16K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

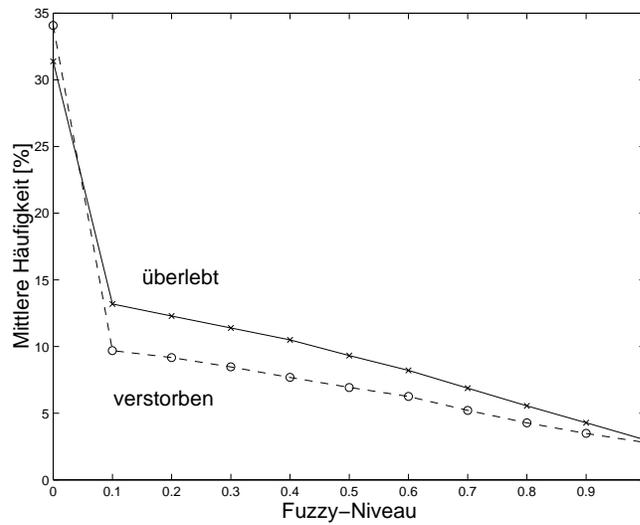


Abbildung 7.8: Häufig16K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

Nr.	Variable	Klasse ü	Klasse v	Summe
1	Herzfrequenz	0.094	0.489	0.583
2	Blutdruck Systolisch	0.281	0.378	0.659
3	Blutdruck Diastolisch	0.477	0.324	0.801
4	Temperatur	0.040	0.028	0.068
5	ZVD	<b>0.511</b>	<b>0.544</b>	<b>1.055</b>
6	O <sub>2</sub> -Sättigung	0.003	0.006	0.009
7	Leukozyten	0.040	0.176	0.216
8	Hämoglobin	0.028	0.081	0.109
9	Hämatokrit	0.235	0.151	0.386
10	Thrombozyten	0.465	0.166	0.631
11	PTT	0.214	0.217	0.431
12	NatriumSerum	0.238	0.238	0.476
13	KaliumSerum	0.085	0.172	0.257
14	KreatininSerum	0.188	0.224	0.412
15	Blutzucker	0.204	0.047	0.251
16	Urinmenge	0.190	0.112	0.302

**Tabelle 7.10:** Häufig16K-Daten: Wichtigkeit der Variablen bez. den Klassen „überlebt“ und „verstorben“ und deren Summe, gemittelt über alle Versuchswiederholungen. Der größte Wert pro Klasse ist fett gedruckt.

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	81.30	4.44	75.18	86.96
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	82.93	2.87	79.41	86.86
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 7.11:** SOFA1K-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse ü	Klasse v
Klasse ü (Training)	38.70	6.58
Klasse v (Training)	12.12	42.61
Klasse ü (Test)	36.46	4.70
Klasse v (Test)	12.37	46.47

**Tabelle 7.12:** SOFA1K-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

### Detailergebnisse SOFA1K:

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 78$ , Ergebnisse siehe Tabellen 7.11 und 7.12 (Klassifikation) und Tabellen 7.13, C.23 und C.24 (Regelgenerierung). Die verwendeten Testdaten kamen in den 5 Testdurchläufen im Mittel zu 48.83% aus Klasse ü und zu 51.17% aus Klasse v. Die Epochenzahl betrug im Mittel 3.8 mit einer Standardabweichung von 1.79 (Minimum 3, Maximum 7).

Um das Klassifikationsergebnis im Hinblick auf die Veränderung der Klassifikationsschwelle besser einschätzen zu können, werden in der Abb. B.5 alle Klassifikationsergebnisse der 5 Versuche aufgetragen, wobei jeweils 10 Punkte eingetragen wurden, also insgesamt 50 Klassifikationen durchgeführt wurden. Für die SOFA1K-Daten erhält man den guten Wert 0.890 (Min.: 0.770, Max.: 0.937).

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	2.4	1.1	1	4
Klasse v	2.4	1.1	1	4

**Tabelle 7.13:** SOFA1K-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

Die Regeln 1 und 3 aus dem 2. der 5 Versuche (Niveau 0.0 und 1.0) werden wiedergegeben:

1) support class **survived** with test frequency **0.860** and test confidence **0.615** from **61** different patients if

**var** *SOFA*  $\leq 12$

core class **survived** with test corefrequency **0.316** and test coreconfidence **0.977** from **24** different patients if

**var** *SOFA*  $\leq 1$

3) support class **deceased** with test frequency **0.735** and test confidence **0.600** from **55** different patients if

**var** *SOFA*  $\leq 17$

core class **deceased** with test corefrequency **0.243** and test coreconfidence **0.909** from **20** different patients if

**var** *SOFA* in (9,16)

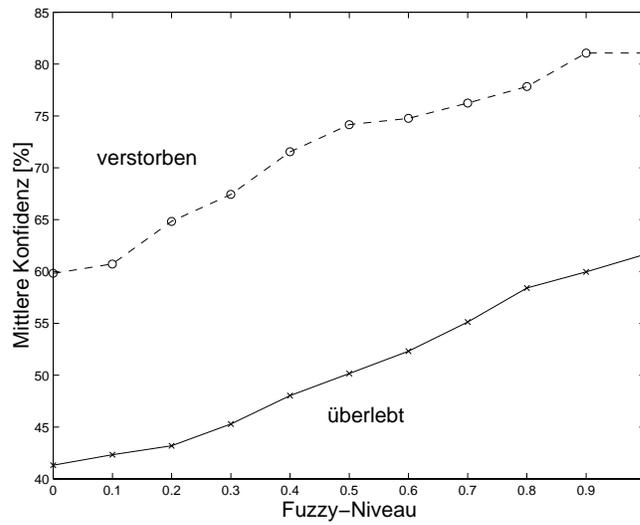
An dieser Stelle sei folgendes angemerkt: Da der SOFA aus 6 Einzelscores summiert wird, sind die hinter diesen Punktskizzen stehenden Regeln, aus den Einzelscores bestehend, naturlich komplexer. Die SOFA-Regeln *erscheinen* also nur einfach. Wurde man z.B. fur den Datensatz Hufig16K die Ausgaben ( $y_1, y_2$ ) berechnen und nach einer Normierung der Ausgabe mit der Differenz zwischen den normierten Ausgaben („Neuronaler-Netze-Score“) wiederum ein neuronales Netz trainieren, so erschienen diese Regeln ebenso einfach wie die des SOFA-Scores.

In der Abb. 7.9 ist die Konfidenz und in der Abb. 7.10 die Hufigkeit in Abhangigkeit vom Fuzzy-Niveau wiedergegeben. Man sieht mustergultige Kurven, so wie man sie erwartet: steigende mittlere Konfidenz, fallende mittlere Hufigkeit. Das resultiert aus der Eindimensionalitat der Score-Daten, also durch das Scoring der Einzelvariablen. Den gleichen Effekt wurde der o.g. „Neuronale-Netze-Score“ erzielen.

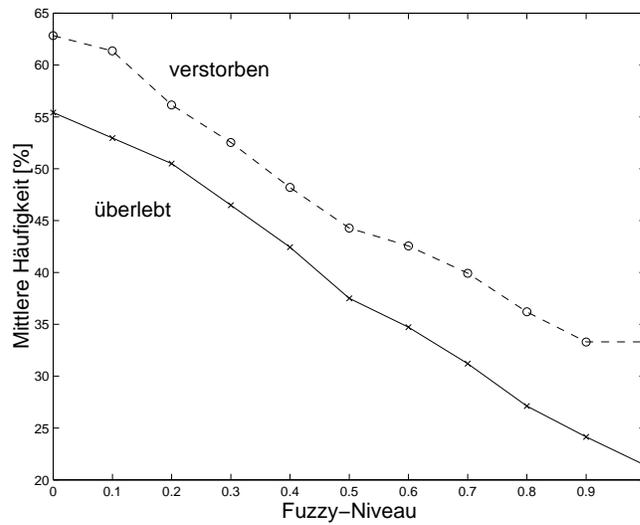
### Detailergebnisse SystThrombo2K:

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 78$ , Ergebnisse siehe Tabellen 7.14 und 7.15 (Klassifikation) und Tabellen 7.16, C.25 und C.26 (Regelgenerierung). Die verwendeten Testdaten kamen in den 5 Testdurchlaufen im Mittel zu 50.47% aus Klasse  $\ddot{u}$  und zu 49.53% aus Klasse  $v$ . Die Epochenzahl betrug im Mittel 4.6 mit einer Standardabweichung von 1.67 (Minimum 3, Maximum 7).

Um das Klassifikationsergebnis im Hinblick auf die Veranderung der Klassifikationsschwelle besser einschatzen zu konnen, werden in der Abb. B.6 alle Klassifikationsergebnisse der 5 Versuche aufgetragen, wobei jeweils 10 Punkte eingetragen wurden,



**Abbildung 7.9:** SOFA1K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 7.10:** SOFA1K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	86.50	4.94	79.87	92.16
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	82.23	2.81	79.22	86.45
Test nicht klassifiziert	0.00	0.00	0.00	0.00

**Tabelle 7.14:** SystThrombo2K-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse ü	Klasse v
Klasse ü (Training)	44.19	6.63
Klasse v (Training)	6.86	42.31
Klasse ü (Test)	42.28	9.58
Klasse v (Test)	8.19	39.95

**Tabelle 7.15:** SystThrombo2K-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

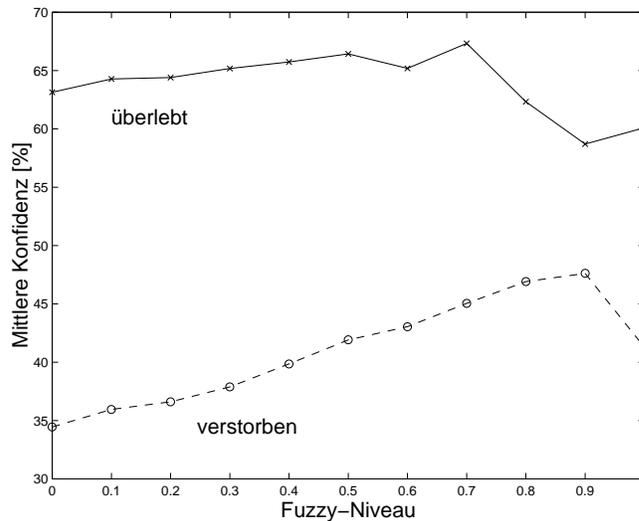
also insgesamt 50 Klassifikationen durchgeführt wurden. Für die SystThrombo2K erhält man den guten Wert 0.878 (Min.: 0.841, Max.: 0.914).

Es folgen zwei interessante Beispielregeln aus dem 3. der 5 Versuche (Regeln 1 und 10):

1) support class **survived** with test frequency **0.252** and test confidence **0.897** from **21** different patients if  
**var** *BlutdruckSystolisch* in (111.67,144.00) and  
**var** *Thrombozyten* in (149.00,417.00)  
core class **survived** with test corefrequency **0.077** and test coreconfidence **1.000** from **10** different patients if  
**var** *BlutdruckSystolisch* in (120.63,138.67) and  
**var** *Thrombozyten* in (233.00,396.00)

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	6.0	1.6	4	8
Klasse v	7.0	0.7	6	8

**Tabelle 7.16:** SystThrombo2K-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).



**Abbildung 7.11:** SystThrombo2K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

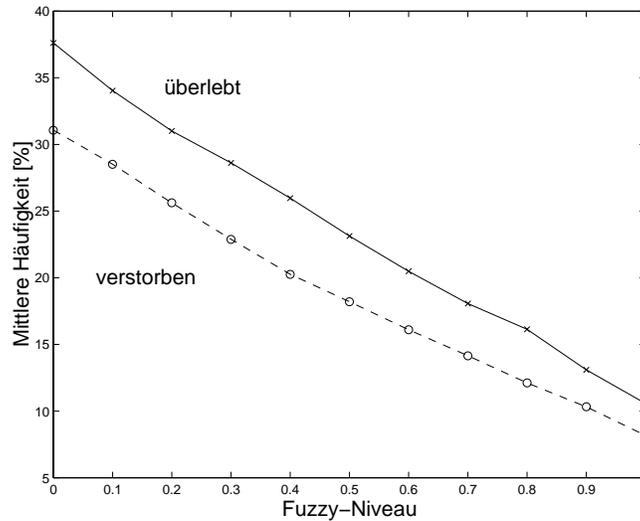
10) support class **deceased** with test frequency **0.348** and test confidence **0.407** from **32** different patients if  
**var** *BlutdruckSystolisch* in (114.50,156.30) and  
**var** *Thrombozyten*  $\leq$  301.00  
 core class **deceased** with test corefrequency **0.071** and test coreconfidence **0.818** from **9** different patients if  
**var** *BlutdruckSystolisch* in (114.50,131.88) and  
**var** *Thrombozyten* in (30.00,149.00)

In der Abb. 7.11 ist die Konfidenz und in der Abb. 7.12 die Häufigkeit in Abhängigkeit vom Fuzzy-Niveau wiedergegeben. Die Konfidenz fällt hier ab dem Fuzzy-Niveau 0.7 bzw. 0.9 für die Klasse *ü* bzw. Klasse *v*.

### Detailergebnisse Wichtig6K:

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 78$ , Ergebnisse siehe Tabellen 7.17 und 7.18 (Klassifikation) und Tabellen 7.19, C.27 und C.28 (Regelgenerierung). Die verwendeten Testdaten kamen in den 5 Testdurchläufen im Mittel zu 49.64% aus Klasse *ü* und zu 50.36% aus Klasse *v*. Die Epochenzahl betrug im Mittel 3.6 mit einer Standardabweichung von 0.55 (Minimum 3, Maximum 4).

Um das Klassifikationsergebnis im Hinblick auf die Veränderung der Klassifikationsschwelle besser einschätzen zu können, werden in der Abb. B.7 alle Klassifikations-



**Abbildung 7.12:** SystThrombo2K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	86.18	3.19	82.39	88.96
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	82.76	2.49	79.88	85.37
Test nicht klassifiziert	0.12	0.27	0.00	0.60

**Tabelle 7.17:** Wichtig6K-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

	Klasse ü	Klasse v
Klasse ü (Training)	45.36	9.23
Klasse v (Training)	4.49	40.82
Klasse ü (Test)	41.65	9.12
Klasse v (Test)	8.00	41.12

**Tabelle 7.18:** Wichtig6K-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	8.2	0.8	7	9
Klasse v	7.8	2.6	5	11

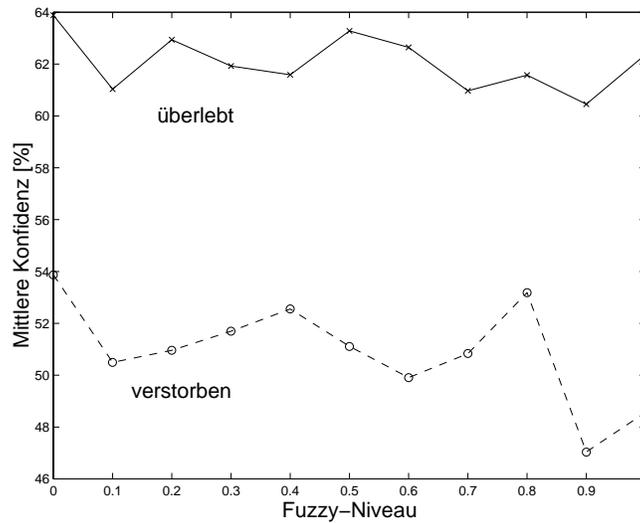
**Tabelle 7.19:** Wichtig6K-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

ergebnisse der 5 Versuche aufgetragen, wobei jeweils 10 Punkte eingetragen wurden, also insgesamt 50 Klassifikationen durchgeführt wurden. Für die Wichtig6K-Daten erhält man den guten Wert 0.882 (Min.: 0.831, Max.: 0.929).

Zwei interessante Beispielregeln (Regeln 1 und 16) aus dem 2. der 5 Versuche (Niveau 0.0 und 1.0):

1) support class **survived** with test frequency **0.451** and test confidence **0.851** from **35** different patients if  
**var** *ZVD* in  $\leq 25.00$  and  
**var** *Thrombozyten*  $\geq 200.99$  and  
**var** *NatriumSerum*  $\leq 147.00$   
core class **survived** with test corefrequency **0.140** and test coreconfidence **0.957** from **14** different patients if  
**var** *Herzfrequenz* in (73.33,104.58) and  
**var** *BlutdruckSystolisch* in (92.73,150.21) and  
**var** *BlutdruckDiastolisch* in (37.27,73.08) and  
**var** *ZVD* in (3.20,10.22) and  
**var** *Thrombozyten* in (216.62,445.00) and  
**var** *NatriumSerum* in (131.00,144.00)

16) support class **deceased** with test frequency **0.341** and test confidence **0.893** from **26** different patients if  
**var** *BlutdruckSystolisch* in  $\leq 121.67$  and  
**var** *ZVD* in  $\geq 4.00$  and  
**var** *Thrombozyten*  $\leq 264.33$   
core class **deceased** with test corefrequency **0.195** and test coreconfidence **1.000** from **21** different patients if  
**var** *Herzfrequenz* in (57.29,158.33) and  
**var** *BlutdruckSystolisch* in (65.59,105.71) and  
**var** *BlutdruckDiastolisch* in (25.67,66.92) and  
**var** *ZVD* in (4.78,26.67) and  
**var** *Thrombozyten* in (8.00,255.58) and  
**var** *NatriumSerum* in (129.33,160.00)



**Abbildung 7.13:** Wichtig6K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

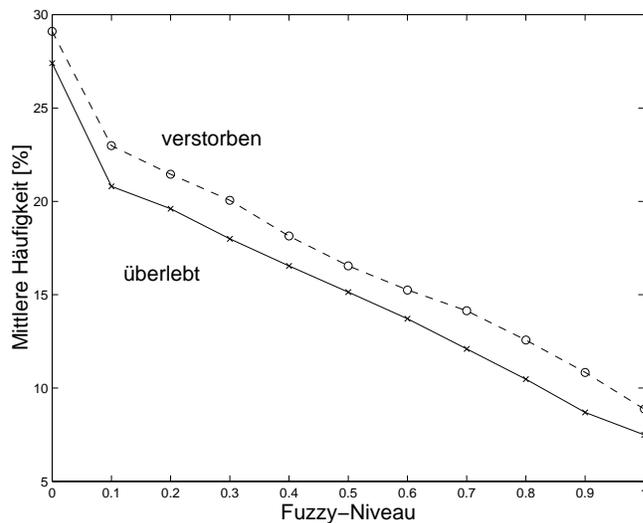
	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	82.37	4.77	74.50	86.70
Training nicht klassifiziert	0.00	0.00	0.00	0.00
Test korrekt	82.99	2.90	80.20	87.50
Test nicht klassifiziert	0.10	0.22	0.00	0.50

**Tabelle 7.20:** SystDiasThrombo3K-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

In der Abb. 7.13 ist die Konfidenz und in der Abb. 7.14 die Häufigkeit in Abhängigkeit vom Fuzzy-Niveau wiedergegeben. Die Konfidenz verändert sich wenig, d.h. kleinere und größere Regeln besitzen in etwa die gleiche Konfidenz. Ab dem Fuzzy-Niveau 0.8 fällt die mittlere Konfidenz für Klasse v etwas ab.

### Detailergebnisse SystDiasThrombo3K:

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 78$ , Ergebnisse siehe Tabellen 7.20 und 7.21 (Klassifikation) und Tabellen 7.22, C.29 und C.30 (Regelgenerierung). Die verwendeten Testdaten kamen in den 5 Testdurchläufen im Mittel zu 50.93% aus Klasse ü und zu 49.07% aus Klasse v. Die Epochenzahl betrug im Mittel 3.8 mit einer Standardabweichung von 1.67 (Minimum 3, Maximum 5).



**Abbildung 7.14:** Wichtig6K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

	Klasse ü	Klasse v
Klasse ü (Training)	42.26	10.86
Klasse v (Training)	6.77	40.12
Klasse ü (Test)	43.80	9.78
Klasse v (Test)	7.13	39.20

**Tabelle 7.21:** SystDiasThrombo3K-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	7.6	1.1	6	9
Klasse v	8.2	1.6	7	11

**Tabelle 7.22:** SystDiasThrombo3K-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

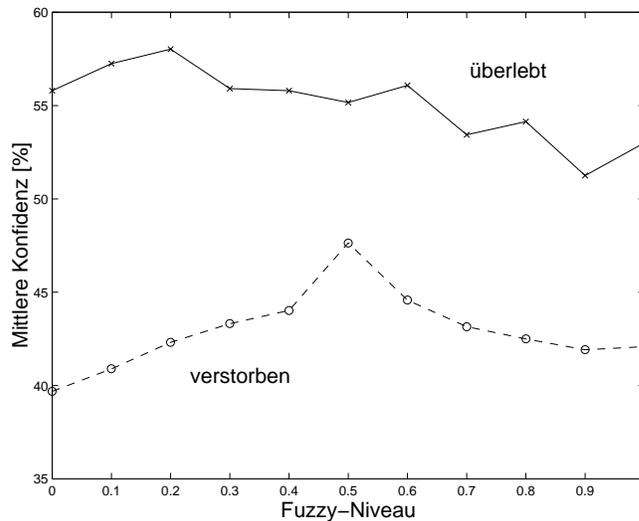
Um das Klassifikationsergebnis im Hinblick auf die Veränderung der Klassifikationsschwelle besser einschätzen zu können, werden in der Abb. B.8 alle Klassifikationsergebnisse der 5 Versuche aufgetragen, wobei jeweils 10 Punkte eingetragen wurden, also insgesamt 50 Klassifikationen durchgeführt wurden. Man erhält für die SystDiasThrombo3K-Daten den guten Wert 0.895 (Min.: 0.849, Max.: 0.929).

Zwei interessante Beispielregeln (Regeln 1 und 9) aus dem 1. der 5 Versuche (Niveau 0.0 und 1.0), die eine sehr gute Häufigkeit und Konfidenz besitzen und auch Daten von vielen Testpatienten abdecken:

1) support class **survived** with test frequency **0.305** and test confidence **0.984** from **30** different patients if  
**var** *BlutdruckSystolisch*  $\geq 111.83$  and  
**var** *BlutdruckDiastolisch*  $\geq 41.67$  and  
**var** *Thrombozyten* in (264.00,700.00)  
core class **survived** with test corefrequency **0.075** and test coreconfidence **1.000** from **11** different patients if  
**var** *BlutdruckSystolisch* in (120.00,150.00) and  
**var** *BlutdruckDiastolisch* in (60.00,97.78) and  
**var** *Thrombozyten* in (321.00,440.00)

9) support class **deceased** with test frequency **0.405** and test confidence **0.914** from **35** different patients if  
**var** *BlutdruckSystolisch*  $\leq 127.50$  and  
**var** *BlutdruckDiastolisch*  $\leq 62.78$ ) and  
**var** *Thrombozyten* in  $\leq 282.00$   
core class **deceased** with test corefrequency **0.215** and test coreconfidence **0.930** from **24** different patients if  
**var** *BlutdruckSystolisch* in (58.00,110.48) and  
**var** *BlutdruckDiastolisch* in (30.88,61.00) and  
**var** *Thrombozyten* in (23.00,261.37)

In der Abb. 7.15 ist die Konfidenz und in der Abb. 7.16 die Häufigkeit in Abhängigkeit vom Fuzzy-Niveau wiedergegeben. Man sieht eine mustergültige Häufigkeits-



**Abbildung 7.15:** SystDiasThrombo3K-Daten: Mittlere Konfidenz der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

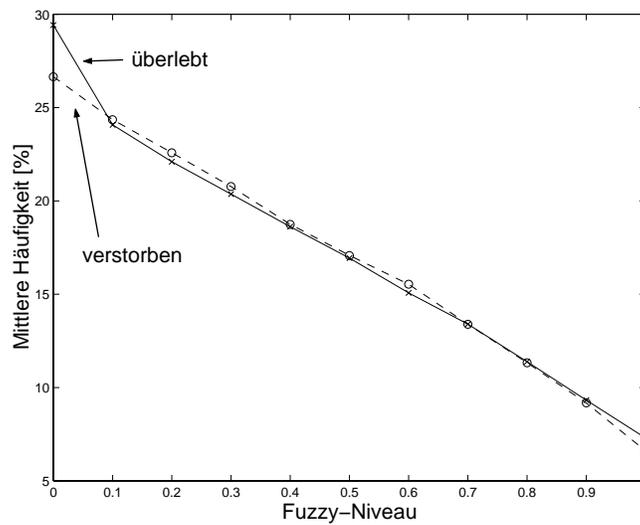
Kurve und eine Konfidenz-Kurve, die zum Niveau 0.5 den besten Wert für die Klasse „verstorben“ hat, für höhere Fuzzy-Niveaus aber nicht mehr stark abfällt.

Die Verteilungen der *optimalen* scharfen Regeln bez. der beiden Klassen sind in den Abbildungen 7.17 und 7.18 angegeben. Als mittlere Konfidenz der optimalen scharfen Regeln ergibt sich für die Klasse  $\bar{u}$  der Wert 72.21% und für die Klasse  $v$  der Wert 60.40%. Diese Werte sind deutlich höher als die maximal möglichen Werte in der Abb. 7.15.

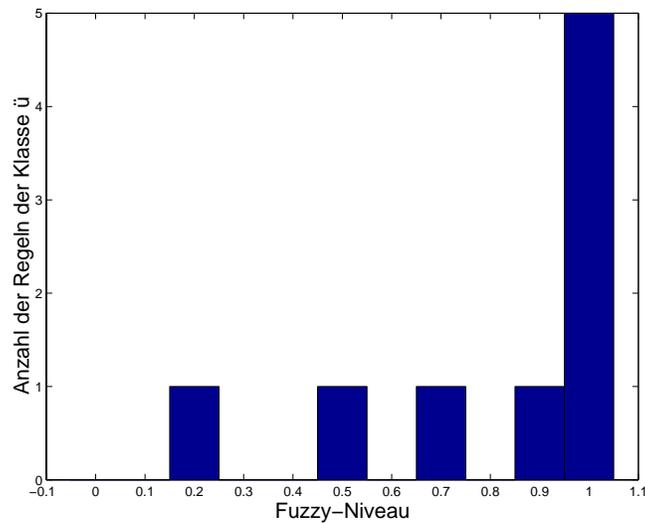
**Ähnlichkeitsberechnung:** Abschließend folgt in der Tabelle 3.7 aus den Ähnlichkeitsberechnungen (siehe Abschnitt 7.23) die Angabe der repräsentativen Regelmengen. Die repräsentative Regelmenge wird im Abschnitt 7.5 für den Aufbau des Alarmsystems verwendet.

Aus den repräsentativen Regelmengen wurden die Beispielregeln dieses Abschnitts entnommen.

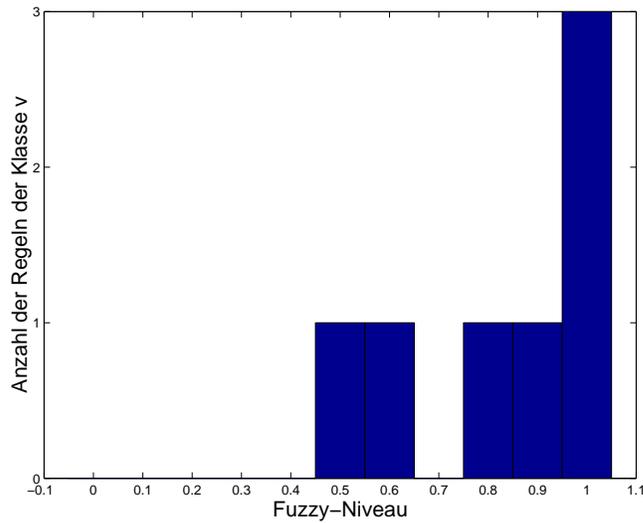
**Vergleich:** Bevor im nächsten Abschnitt die symbolischen Klinikdaten analysiert werden, soll an dieser Stelle noch ein kurzes Fazit aus den Ergebnissen der metrischen Vorstudien und Klinikdaten gezogen werden. Die Klassifikationsergebnisse (ROC-Flächeninhalt) als auch die Regelperformance (Konfidenz, Häufigkeit) auf den Klinikdaten sind besser. Sie sind gut genug, um ein Alarmsystem sinnvoll aufbauen zu können (Abschnitt 7.5), insbesondere durch die Berücksichtigung der Zeitabhängigkeit der Analysen der Klinikdaten. Die Klinikdaten hatten weniger fehlende Werte und insgesamt eine bessere Dokumentationsqualität.



**Abbildung 7.16:** SystDiasThrombo3K-Daten: Mittlere Häufigkeit der Regeln pro Klasse in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 7.17:** SystDiasThrombo3K-Daten: Verteilung der optimalen scharfen Regeln der Klasse ü in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).



**Abbildung 7.18:** SystDiasThrombo3K-Daten: Verteilung der optimalen scharfen Regeln der Klasse  $v$  in Abhängigkeit vom Fuzzy-Niveau (in 0.1-Schritten).

Datensatz	#Versuche	„best“ $j$	$Sim(R_j)$	„worst“ $k$	$Sim(R_k)$
Häufig16K	5	1	0.4711	2	0.4452
SOFA1K	5	2	0.7883	5	0.5483
SystThrombo2K	5	3	0.8171	4	0.7949
Wichtig6K	5	2	0.6453	5	0.6184
SystDiasThrombo3K	5	1	0.7989	4	0.7641

**Tabelle 7.23:** Repräsentative Regelmengen. Wiedergegeben ist die Anzahl der Versuche ( $\#$ Versuche), der Index  $j$  der repräsentativen Regelmenge  $R_j$  und der Wert  $Sim(R_j)$ . Angegeben ist auch der Index  $k$  der ungeeignetsten Regelmenge, zusammen mit  $Sim(R_k) = \min_l \{Sim(R_l)\}$ .

Die Unterschiede zwischen Klinikdaten und Vorstudien­daten liefern einen deutlichen Hinweis auf die Abhängigkeit der Ergebnisse von der Datenqualität. Insgesamt ist es gelungen, sinnvolle Ergebnisse auf strukturell schwierigen medizinischen Daten zu erzielen, wenngleich die mittlere Konfidenz zu verschiedenen Fuzzy-Niveaus immer noch auf eine etwas zu geringe Patientenzahl schließen lässt.

## 7.3 Erzeugung von Generalisierungsregeln

In den nächsten beiden Abschnitten werden die Verfahren aus dem Kapitel 4, in Analogie zu der Anwendung auf den ASK-Daten im Kapitel 6, auf die symbolischen Klinikdaten aller Kliniken angewendet. Als erstes werden Regeln mit allen 145 Items generiert; anschließend wird mit den wichtigsten Items erneut eine Regelgenerierung durchgeführt.

### 7.3.1 Generalisierung mit allen 145 Items

Die symbolischen Klinikdaten beinhalten neben binären Medikament- bzw. Therapievariablen („ja/nein“) auch Operationen und Diagnosen („ja“). Da pro Patient nur wenige Operationen bzw. Diagnosen aus einer sehr großen Anzahl möglicher Operationen bzw. Diagnosen vorkommen, wurde auf die Aufnahme von Verneinungen für diese symbolischen Variablen verzichtet. Für jeden der 138 septischen Schock-Patienten (68 überlebt, 70 verstorben, d.h. 49.28% überlebt, 50.72% verstorben) wurde ein Itemset aus den zugehörigen Patientendaten gebildet. Fehlende Werte spielen hier keine Rolle, da von einer vollständigen Dokumentation (bis auf unbekannte Fehler) ausgegangen werden kann. Ein „ja“-Item wurde in das Itemset für eine Variable aufgenommen, wenn diese Variable während des Intensiv­aufenthalts mind. einmal auf „ja“ gesetzt war. Waren für eine Variable zusätzlich Klassen vorhanden, so wurde für die Klasse ein Item in das Itemset aufgenommen, wenn diese Klasse während des Intensiv­aufenthalts mind. einmal vorgekommen ist. Hatten die Medikament- bzw. Therapievariablen während des Intensiv­aufenthalts kein „ja“, so wurde ein Item „nein“ aufgenommen.

Anhand der 138 Patienten (gleich Anzahl der Itemsets) wurden hier drei Versuchswiederholungen durchgeführt. Im Mittel bestand jedes Itemset aus 55.0 Items (Minimum: 45, Maximum: 71). Insgesamt waren 145 verschiedene Items kodiert. Es wurden die Heuristiken 1. und 4. aus dem Abschnitt 4.6 angewendet sowie die Generalisierungsindices verwendet. Für das erste Level wurden 50% der Daten zum Training verwendet, die anderen 50% zum Testen, u.a zur Bestimmung der Klassifikationsleistung. Es lagen jeweils folgende Einstellungen vor:  $\max_{new} = 4$ ,  $\min_{freq} = 0.32$  (3. Versuch: 0.23) sowohl für die Trainings- als auch die Test-Itemsets für die Klasse „überlebt“ (Klasse  $\ddot{u}$ ) und  $\min_{freq} = 0.17$  (2. Versuch: 0.20, 3. Versuch: 0.21) für die Klasse „verstorben“ (Klasse  $v$ ). Für die Testmenge wurde die minimale Konfidenz

auf  $min_{conf} = 0.80$  (3. Versuch: 0.77) für die Klasse „überlebt“ und  $min_{conf} = 0.76$  (2. Versuch: 0.79, 3. Versuch: 0.78) für die Klasse „verstorben“ gesetzt. Die erzeugten Regeln wurden nach diesen Kriterien gefiltert. Dabei dienen diese Versuche vor allem der heuristischen Regelgenerierung, um mit Hilfe der Wichtigkeit eine Reduzierung der Items zu erreichen. Anschließend werden dann mit den wichtigsten Items erneut Regeln generiert.

In der Tab. 7.24 sind die Ergebnisse sowie der Mittelwert aller drei Versuche zum Vergleich wiedergegeben.

Die Fenstergröße zur Berechnung von  $G^{(in)}$  wurde auf 15, die Abbruchschwelle für das Unterschreiten von  $G^{(in)}$  wurde auf 0.4 und die Abbruchschwelle für das Unterschreiten von  $G^{(out)}$  wurde auf 0.1 gesetzt. In der Tab. 7.25 sind die äußeren Generalisierungsindices dokumentiert, wenngleich bei diesen Versuchen bereits generell nach dem dritten Level abgebrochen wurde. Man sieht aber, dass es evtl. sinnvoll wäre, weitere Level zu berücksichtigen, wenn man an dieser Stelle bereits ein Feintuning der Regeln vornehmen wollte. Hier sollen zuerst die wichtigen Items identifiziert werden.

Der Flächeninhalt unter der ROC-Kurve (pro Versuch 10 Klassifikationen) betrug im 1. Versuch 0.701, im 2. Versuch 0.748 und im 3. Versuch 0.723, im Mittel 0.724.

In der Tab. 7.26 wird die Wichtigkeit derjenigen Itemsets wiedergegeben (Addition aus den 3 Versuchen pro Klasse), für die der Unterschied zwischen den Klassen mind. 0.095 betragen hat. Mit diesen Items wird erneut eine Regelgenerierung durchgeführt. Die Wichtigkeiten der anderen Items unterscheiden sich um weniger als 0.095. 99 der 145 Items besitzen für beide Klassen die Wichtigkeit 0. Darunter sind (bedauerlicherweise) auch alle Diagnosen und Operationen zu finden; mit Hilfe der Medikamente/Therapien konnten die deutlich besseren Regeln erzeugt werden.

Bei der Betrachtung der Tab. 7.26 fällt auf, dass sowohl „Kristalloide ja“ als auch „Kristalloide nein“ für die Klasse „überlebt“ die höheren Werte hat. So etwas ist prinzipiell möglich, da die verschiedenen Items in Kombination mit anderen Items verschiedene Regeln bilden können, die für die Klasse „überlebt“ eine höhere Konfidenz bzw. Häufigkeit besitzen. Es fällt noch auf, dass nur eine einzige Unterklasse Berücksichtigung findet; anscheinend ist eine Einteilung in Unterklassen für nur 138 Patienten nicht sinnvoll gewesen (zu niedrige Fallzahl). Die Unterklasse Opiode gehört zur Oberklasse Analgetika, die sowohl für die Klasse „überlebt“ als auch für die Klasse „verstorben“ die höchsten aller Wichtigkeitswerte hat, d.h. dass erst die Kombination anderer Items mit Analgetika die Klassenzugehörigkeit bestimmt.

Wie bereits im Abschnitt 4.9 vorgestellt, wird die repräsentative Regelmenge bestimmt (Parametereinstellungen:  $\tau = 0.2$ ,  $\kappa_1 = 0.2$  für die Häufigkeit auf den Test-

	Versuch 1	Versuch 2	Versuch 3	Mittelwert
Anzahl der Regeln	34	31	33	32.7
Regeln (Kl. ü)	9	24	24	19.0
Regeln (Kl. v)	25	7	9	13.7
Länge der Regeln (Kl. ü)	14.5	16.0	17.6	16.0
Länge der Regeln (Kl. v)	15.1	16.7	16.9	16.2
korrekt klassifiziert [%]	55.07	60.87	52.17	56.04
falsch klassifiziert [%]	30.44	20.29	14.49	21.74
nicht klassifiziert [%]	14.49	18.84	33.33	22.22
Spezifität [%]	66.67	72.00	76.00	71.56
Sensitivität [%]	62.07	77.42	80.95	73.48
Test-Konfidenz (Kl. ü) [%]	83.94	82.16	80.28	82.13
Test-Konfidenz (Kl. v) [%]	81.63	81.63	81.74	81.67
gew. Test-Konf. (Kl. ü) [%]	84.32	78.99	79.82	81.04
gew. Test-Konf. (Kl. v) [%]	81.20	84.48	82.17	82.62
Test-Häufigkeit (Kl. ü) [%]	35.99	36.66	25.54	32.73
Test-Häufigkeit (Kl. v) [%]	19.10	22.36	23.67	21.71
Frühzeitigkeit (Kl. ü) [Tage]	9.52	8.61	8.88	9.00
Frühzeitigkeit (Kl. v) [Tage]	6.80	6.27	8.85	7.31
Lerndauer [min.]	22	25	21	22.7

**Tabelle 7.24:** Klinikdaten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln. Die Sensitivität und die Spezifität wurde ohne die nicht klassifizierten Itemsets berechnet. Die Test-Konfidenz, die gewichtete (Abk.: gew.) Test-Konfidenz, die Häufigkeit und die Frühzeitigkeit ist pro Versuch als Mittel über die Anzahl der erzeugten Regeln angegeben. Bei der (gew.) Test-Konfidenz wurden nur Werte größer als die im Text genannten Schwellen für den Mittelwert berücksichtigt, da das Itemset sonst keine Regel dieser Klasse darstellt. Regeln der Klasse  $c$  (Zeile 2 und 3) sind Regeln, deren (gew.) Test-Konfidenz für diese Klasse maximal ist. Dasselbe gilt für die Länge der Regeln, die gemittelt pro Versuch angegeben ist.

Level	neu erzeugt	möglich	$G^{(out)}$
1	69 / 69 / 69	- / - / -	- / - / -
2	256 / 256 / 256	276 / 276 / 276	0.928 / 0.928 / 0.928
3	972 / 974 / 973	1024 / 1024 / 1024	0.949 / 0.951 / 0.950

**Tabelle 7.25:** Klinikdaten: Neu erzeugte, möglich zu erzeugende Itemsets und äußere Generalisierungsindices. Versuch1/Versuch2/Versuch3.

Item-Nr.	Item-Name	überlebt	verstorben
1	Schleifendiuretika ja	0.211	0.625
2	Pentaglobin nein	1.242	0.372
3	Kristalloide ja	1.309	0.729
4	Kristalloide nein	0.951	0.287
5	EK ja	0.022	0.349
6	FFP ja	0.020	0.439
7	AdrenalinPerfusor nein	0.908	0.096
8	DobutaminPerfusor ja	0.011	0.149
9	Haemofiltration nein	1.138	0.157
10	ParenteraleErnaehrung ja	1.300	0.553
11	Lungeninfiltration ja	0.704	0.047
12	Analgetika ja	<b>1.319</b>	<b>0.947</b>
13	Analgetika UK Opioide	<b>1.319</b>	<b>0.947</b>
14	Antibiotika ja	1.315	0.875
15	Antihypertonika ja	0.105	0.008
16	Antihypotonika nein	0.078	0.178
17	Antikoagulation ja	1.221	0.503
18	Antimykotika nein	0.008	0.148
19	Antipyretika nein	0.007	0.129
20	Natriumbicarbonat(Einfuhr) ja	0.005	0.203
21	Natriumbicarbonat(Einfuhr) nein	0.153	0.003
22	Diuretika nein	0.253	0.107
23	ImmunglobulinePolyglobuline nein	0.904	0.533
24	Albumin5(Einfuhr) nein	0.507	0.408
25	PPSB nein	0.444	0.295
26	einzelne Gerinnungsfaktoren nein	0.864	0.673
27	SedativaNarkotika ja	0.311	0.543
28	Beatmungsform ja	0.387	0.288
29	Dialyse nein	0.297	0.765
30	Katheter ja	0.770	0.226
31	EnteraleErnaehrung ja	0.361	0.023
32	EnteraleErnaehrung nein	0.015	0.282
33	Albumin20(Einfuhr) nein	0.042	0.140
34	AdrenalinEinmalgabe nein	1.237	0.318
35	NoradrenalinEinmalgabe nein	1.099	0.510

**Tabelle 7.26:** Klinikdaten: Summierte Wichtigkeit der Items (additiv aus den drei Versuchen gebildet). UK ist Abk. für Unterklasse. Die höchsten Werte pro Klasse sind fett hervorgehoben.

Level	neu erzeugt	möglich	$G^{(out)}$
1	68 / 69 / 68	- / - / -	- / - / -
2	252 / 256 / 252	272 / 276 / 272	0.927 / 0.928 / 0.927
3	945 / 968 / 944	1008 / 1024 / 1008	0.938 / 0.945 / 0.937

**Tabelle 7.27:** Klinikdaten (35 wichtigste Items): Neu erzeugte, möglich zu erzeugende Itemsets und äußere Generalisierungsindices. Versuch1/Versuch2/Versuch3.

daten,  $\kappa_2 = \kappa_3 = 0.2$  für die Konfidenz auf den Testdaten für beide Klassen). Man erhält:  $Sim(R_1) = 0.6717$ ,  $Sim(R_2) = 0.6936$  und  $Sim(R_3) = 0.6743$ . Da hier nur drei Regelmengen vorliegen und die Werte nahe beieinander liegen, ist eine Auszeichnung einer Regelmenge nicht unbedingt sinnvoll, aber in diesem Fall wäre die Regelmenge 2 die repräsentative.

Auf die Frühzeitigkeit der Regeln und die Ähnlichkeit der Regeln wird hier nicht weiter im Detail eingegangen. Dies geschieht im Anschluss an den Versuch mit den wichtigsten Items; dort werden dann auch Regelbeispiele genannt. Zur Frühzeitigkeit sei hier lediglich angemerkt, dass hier die Frühzeitigkeit für die Regeln der Klasse „überlebt“ höher ist im Unterschied zu den ASK-Daten, wo die Frühzeitigkeit für die Regeln der Klasse „verstorben“ höher war.

### 7.3.2 Generalisierung mit den wichtigsten 35 Items

Ausgehend von der Tab. 7.26 wird mit den dort genannten 35 von 145 Items (24.1% der Items) erneut eine Regelgenerierung durchgeführt.

Anhand der 138 Patienten (gleich Anzahl der Itemsets) wurden hier drei Versuchswiederholungen durchgeführt. Im Mittel bestand jedes Itemset aus 26.6 Items (Minimum: 21, Maximum: 31). Es wurden die Heuristiken 1. und 4. aus dem Abschnitt 4.6 angewendet sowie die Generalisierungsindices. Für das erste Level wurden 50% der Daten zum Training verwendet, die anderen 50% zum Testen, u.a zur Bestimmung der Klassifikationsleistung. Es lagen jeweils folgende Einstellungen vor:  $\max_{new} = 4$ ,  $\min_{freq} = 0.27$  (2. Versuch: 0.28, 3. Versuch: 0.20) sowohl für die Trainings- als auch für die Test-Itemsets für die Klasse „überlebt“ (Klasse  $\ddot{u}$ ) und  $\min_{freq} = 0.23$  (2. Versuch: 0.22, 3. Versuch: 0.24) für die Klasse „verstorben“ (Klasse  $v$ ). Für die Testmenge wurde die minimale Konfidenz auf  $\min_{conf} = 0.80$  (2. Versuch: 0.82, 3. Versuch: 0.81) für die Klasse „überlebt“ und  $\min_{conf} = 0.82$  (2. Versuch: 0.80, 3. Versuch: 0.83) für die Klasse „verstorben“ gesetzt. Die erzeugten Regeln wurden nach diesen Kriterien gefiltert.

	Versuch 1	Versuch 2	Versuch 3	Mittelwert
Anzahl der Regeln	35	31	29	31.7
Regeln (Kl. ü)	15	21	11	15.7
Regeln (Kl. v)	20	10	18	16.0
Länge der Regeln (Kl. ü)	13.0	15.0	15.6	14.5
Länge der Regeln (Kl. v)	13.2	14.9	15.0	14.4
korrekt klassifiziert [%]	63.77	52.17	55.07	57.00
falsch klassifiziert [%]	15.94	24.64	27.54	22.71
nicht klassifiziert [%]	20.29	23.19	17.39	20.29
Spezifität [%]	96.97	65.63	51.52	71.37
Sensitivität [%]	54.55	71.43	87.50	71.16
Test-Konfidenz (Kl. ü) [%]	80.47	85.08	85.38	83.64
Test-Konfidenz (Kl. v) [%]	84.58	84.94	84.31	84.61
gew. Test-Konf. (Kl. ü) [%]	85.04	85.44	87.73	86.07
gew. Test-Konf. (Kl. v) [%]	79.94	84.57	81.43	81.98
Test-Häufigkeit (Kl. ü) [%]	29.18	30.02	22.27	27.16
Test-Häufigkeit (Kl. v) [%]	26.59	25.94	27.21	26.58
Frühzeitigkeit [Tage] (Kl. ü)	11.32	11.48	10.13	10.98
Frühzeitigkeit [Tage] (Kl. v)	7.21	8.07	9.14	8.14
Lerndauer [min.]	3	4	3	3.7

**Tabelle 7.28:** Klinikdaten: Ergebnis-Kennzahlen der erzeugten Generalisierungsregeln unter Berücksichtigung der Wichtigkeit. Die weiteren Erklärungen sind identisch zu denen in der Tab. 7.24.

Die Fenstergröße zur Berechnung von  $G^{(in)}$  wurde auf 15, die Abbruchschwelle für das Unterschreiten von  $G^{(in)}$  wurde auf 0.5 und die Abbruchschwelle für das Unterschreiten von  $G^{(out)}$  wurde auf 0.1 gesetzt. In der Tabelle 7.27 sind die äußeren Generalisierungsindices dokumentiert, wengleich bei diesen Versuchen bereits generell nach dem dritten Level abgebrochen wurde. Weitere Versuche zeigten, dass mit keiner prinzipiellen Verbesserung mehr zu rechnen ist, wenn man weitere Level zulässt, so dass der Abbruch gerechtfertigt ist. Weitere Level wären mit einer deutlich höheren Laufzeit verbunden, wengleich die Regeln etwas kürzer werden könnten.

In der Tab. 7.28 sind die Ergebnisse sowie der Mittelwert aller drei Versuche zum Vergleich wiedergegeben. Der Flächeninhalt unter der ROC-Kurve (bei 10 Klassifikationen) betrug im 1. Versuch 0.747, im 2. Versuch 0.818 und im 3. Versuch 0.718, im Mittel 0.761. Das ist ein etwas besseres Ergebnis als mit allen 145 Items.

In der Tab. 7.29 wird die Wichtigkeit derjenigen Items wiedergegeben (Addition aus den 3 Versuchen pro Klasse), deren Wichtigkeit für die erste Versuchsserie bereits in der Tab. 7.26 dokumentiert wurde.

Item-Nr.	Item-Name	überlebt	verstorben
1	Schleifendiuretika ja	0.834	0.290
2	Pentaglobin nein	1.358	0.224
3	Kristalloide ja	0.886	0.920
4	Kristalloide nein	0	0
5	EK ja	0.041	0.688
6	FFP ja	0.012	0.211
7	AdrenalinPerfusor nein	0.447	0.180
8	DobutaminPerfusor ja	0.031	0.589
9	Haemofiltration nein	1.348	0.038
10	ParenteraleErnaehrung ja	1.023	1.240
11	Lungeninfiltration ja	0	0
12	Analgetika ja	<b>1.415</b>	<b>1.302</b>
13	Analgetika UK Opioide	<b>1.415</b>	<b>1.302</b>
14	Antibiotika ja	1.188	1.005
15	Antihypertonika ja	0.060	0.002
16	Antihypotonika nein	0.027	0.113
17	Antikoagulation ja	0.927	0.845
18	Antimykotika nein	0.016	0.293
19	Antipyretika nein	0.003	0.039
20	Natriumbicarbonat(Einfuhr) ja	0	0
21	Natriumbicarbonat(Einfuhr) nein	0.889	0.025
22	Diuretika nein	0.108	0.089
23	ImmunglobulinePolyglobuline nein	1.102	0.513
24	Albumin5(Einfuhr) nein	0.802	0.588
25	PPSB nein	0.965	0.593
26	einzelne Gerinnungsfaktoren nein	0.319	1.222
27	SedativaNarkotika ja	0.846	0.946
28	Beatmungsform ja	0.343	0.319
29	Dialyse nein	1.033	0.548
30	Katheter ja	0.631	0.218
31	EnteraleErnaehrung ja	0.558	0.015
32	EnteraleErnaehrung nein	0.032	0.655
33	Albumin20(Einfuhr) nein	0.057	0.532
34	AdrenalinEinmalgabe nein	1.348	0.038
35	NoradrenalinEinmalgabe nein	1.287	0.974

**Tabelle 7.29:** Klinikdaten: Summierte Wichtigkeit der 35 wichtigsten Items (additiv aus den drei Versuchen gebildet). UK ist Abk. für Unterklasse. Die höchsten Werte pro Klasse sind fett hervorgehoben.

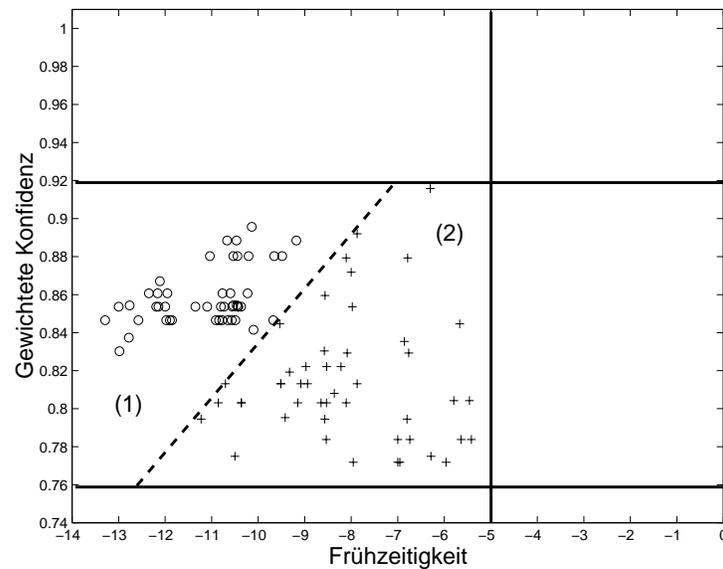
Anhand der Tab. 7.29 kann man versuchen, die interessantesten Indikatoren (als Einzel-Items) für die beiden Klassen zu extrahieren. Ausgewählt wurden hierzu Items, die für eine Klasse eine summierte Wichtigkeit  $< 0.1$  hatten und für die andere Klasse eine  $> 0.5$ . Man erhält folgende Indikatoren:

1. **überlebt:** Item 9 Hämofiltration nein,
2. **überlebt:** Item 21 Natriumbicarbonat(Einfuhr) nein,
3. **überlebt:** Item 34 AdrenalinEinmalgabe nein,
4. **verstorben:** Item 5 EK ja,
5. **verstorben:** Item 8 DobutaminPerfusor ja,
6. **verstorben:** Item 32 EnteraleErnährung nein,
7. **verstorben:** Item 33 Albumin20(Einfuhr) nein.

In der Abb. 7.19 ist die Frühzeitigkeit der Regeln aller drei erzeugten Regelmengen in Abhängigkeit zur gewichteten Konfidenz aufgetragen. Keine der Warnungen tritt im Mittel zum erstenmal später als 5 Tage vor der Entlassung bzw. dem Versterben auf. Die Regeln haben eine gewichtete Konfidenz, die zwischen 76% und 92% liegt. Die Regeln für die Klasse „überlebt“ treten früher auf als die der Klasse „verstorben“. Das ist ein deutlich anderes Ergebnis als dasjenige, das mit den ASK-Daten im Abschnitt 6.3 erzielt wurde. Anscheinend sorgte dort die größere Anzahl Überlebender (evtl. ohne septischen Schock) für eine andere Struktur der Frühzeitigkeit der Regeln.

Interessant ist hier noch die Tatsache, dass die Regeln aus den drei Regelmengen linear separierbar sind, wenn man die gewichtete Konfidenz in Abhängigkeit der Frühzeitigkeit betrachtet (gestrichelte Linie in der Abb. 7.19).

Für ein Frühwarnsystem hätte die Struktur der Frühwarnungen folgende Konsequenzen: Erhält der Arzt in den ersten zwei bis drei Tagen nach der Einlieferung Entwarnungen, so ist der Patient eher unkritisch in Hinblick auf das Versterben. Erhält der Arzt Warnungen nach einigen Tagen Liegedauer, so deutet das auf einen eher sehr kritischen Patienten hin. Nicht vergessen werden darf allerdings, dass die Regeln keine 100%-ige Sicherheit garantieren und das einige Einzelfälle nicht durch die Regeln abgedeckt sein können. Da die Medikamenten- und Therapievariablen aber bereits die Behandlung des Arztes wiedergeben, kann man eigentlich nicht mehr von einer „Frühwarnung“ sprechen, sondern es wird die Behandlung des Arztes klassifiziert.



**Abbildung 7.19:** Klinikdaten: Gewichtete Konfidenz in Abhängigkeit von der Frühzeitigkeit für beide Klassen gemeinsam.  $-x$  [Tage] bedeutet  $x$  Tage vor der Entlassung bzw. vor dem Versterben. Es sind zwei Gruppen erkennbar: (1) Entwarnungen für „überlebt“ (Kreis) und (2) Warnungen für „verstorben“ (Kreuz).

Wie bereits im Abschnitt 4.9 vorgestellt, wird die repräsentative Regelmengung bestimmt (Parametereinstellungen:  $\tau = 0.2$ ,  $\kappa_1 = 0.2$  für die Häufigkeit auf den Testdaten,  $\kappa_2 = \kappa_3 = 0.2$  für die Konfidenz auf den Testdaten für beide Klassen). Man erhält:  $Sim(R_1) = 0.7259$ ,  $Sim(R_2) = 0.7240$  und  $Sim(R_3) = 0.7182$ . Da hier nur drei Regelmengungen vorliegen und die Werte nahe beieinander liegen, ist eine Auszeichnung einer Regelmengung nicht unbedingt sinnvoll, aber in diesem Fall wäre die Regelmengung 1 die repräsentative.

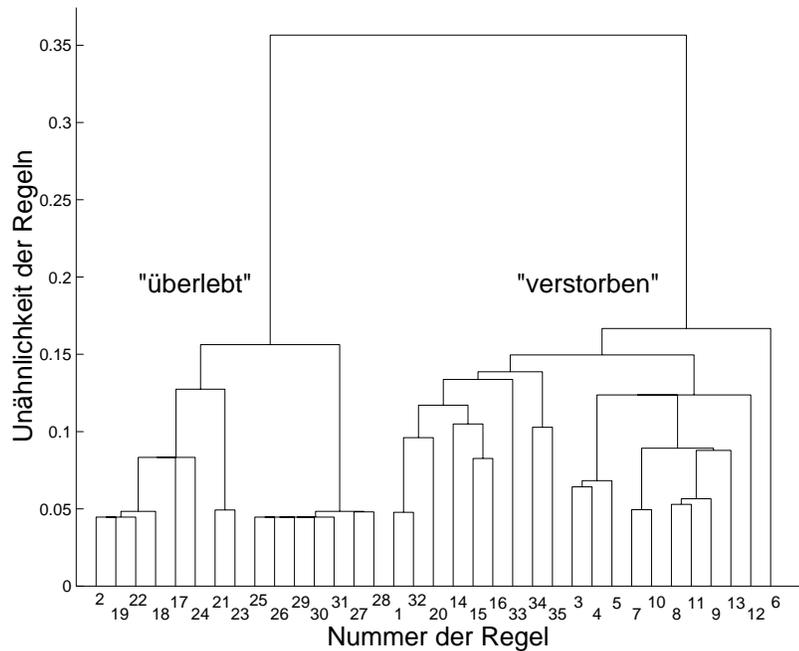
Die (Un-)Ähnlichkeit der Regeln aus der Regelmengung 1 ist in dem Dendrogramm der Abb. 7.20 zu sehen. Es fällt auf, dass die Regeln für die Klasse „verstorben“ (Regeln im rechten Ast) zueinander unähnlicher sind als die Regeln der Klasse „überlebt“ (Regeln im linken Ast). Das lässt auf ein individuelleres Verhalten der verstorbenen Patienten schließen.

Ausgehend von der Abb. 7.20 werden einige markante Regelbeispiele genannt (die Regeln 2, 23 und 25 für „überlebt“ und 1, 15, 34, 3, 7, 12 und 6 für „verstorben“).<sup>5</sup>

Es folgen zuerst die Regelbeispiele der Klasse „überlebt“:

2) „if Schleifendiuretika = ja **and** Pentaglobin = nein **and** Kristalloide = ja **and** Haemofiltration = nein **and** Analgetika = ja **and** Analgetika UK Opiode **and** An-

<sup>5</sup>Die Konfidenz- und Häufigkeitswerte wurden wiederum auf den Testdaten berechnet.



**Abbildung 7.20:** Klinikdaten: Dendrogramm für die Unähnlichkeit der erzeugten Generalisierungsregeln.

tibiotika = ja **and** Natriumbicarbonat(Einfuhr) = nein **and** ImmunglobulinePolyglobuline = nein **and** Dialyse = nein **and** Albumin5(Einfuhr) = nein **and** PPSB = nein **and** AdrenalinEinmalgabe = nein **and** NoradrenalinEinmalgabe = nein **then** Klasse überlebt **with** confidence 80.00% **and** frequency 28.99%<sup>6</sup>

23) „**if** Schleifendiuretika = ja **and** Pentaglobin = nein **and** Kristalloide = ja **and** Haemofiltration = nein **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opiode **and** Antibiotika = ja **and** Antihypertonika = ja **and** Antikoagulation = ja **and** PPSB = nein **and** SedativaNarkotika = ja **and** Dialyse = nein **and** Katheter = ja **and** AdrenalinEinmalgabe = nein **and** NoradrenalinEinmalgabe = nein **then** Klasse überlebt **with** confidence 84.21% **and** frequency 27.54%“

25) „**if** Pentaglobin = nein **and** Kristalloide = ja **and** Haemofiltration = nein **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opiode **and** Antikoagulation = ja **and** ImmunglobulinePolyglobuline = nein **and** SedativaNarkotika = ja **and** Beatmungsform = ja **and** Katheter = ja **and** EnteraleErnaehrung = ja **and** AdrenalinEinmalgabe = nein **and** NoradrenalinEinmalgabe = nein **then** Klasse überlebt **with** confidence 80.00% **and** frequency 28.99%“

<sup>6</sup>„Analgetika = ja **and** Analgetika UK Opiode“ kann man zu „Analgetika = UK Opiode“ zusammenfassen. UK steht für Unterklasse.

Die Regelbeispiele der Klasse „verstorben“ lauten:

1) „**if** Kristalloide = ja **and** EK = ja **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opioiden **and** Antibiotika = ja **and** Antikoagulation = ja **and** ImmunglobulinePolyglobuline = nein **and** Albumin5(Einfuhr) = nein **and** PPSB = nein **and** einzelne Gerinnungsfaktoren = nein **and** SedativaNarkotika = ja **and** Beatmungsform = ja **and** EnteraleErnaehrung = nein **and** NoradrenalinEinmalgabe = nein **then** Klasse verstorben **with** confidence 85.00% **and** frequency 28.99%“

15) „**if** Schleifendiuretika = ja **and** Kristalloide = ja **and** EK = ja **and** DobutaminPerfusor = ja **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opioiden **and** Antibiotika = ja **and** Antihypotonika = nein **and** Antikoagulation = ja **and** Albumin5(Einfuhr) = nein **and** PPSB = nein **and** einzelne Gerinnungsfaktoren = nein **and** SedativaNarkotika = ja **and** Dialyse = nein **and** Albumin20(Einfuhr) = nein **then** Klasse verstorben **with** confidence 83.33% **and** frequency 26.09%“

34) „**if** Pentaglobin = nein **and** Kristalloide = ja **and** EK = ja **and** FFP = ja **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opioiden **and** Antibiotika = ja **and** ImmunglobulinePolyglobuline = nein **and** Albumin5(Einfuhr) = nein **and** SedativaNarkotika = ja **and** Dialyse = nein **and** EnteraleErnaehrung = nein **and** NoradrenalinEinmalgabe = nein **then** Klasse verstorben **with** confidence 82.35% **and** frequency 24.64%“

3) „**if** EK = ja **and** AdrenalinPerfusor = nein **and** DobutaminPerfusor = ja **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opioiden **and** Antikoagulation = ja **and** Diuretika = nein **and** PPSB = nein **and** einzelne Gerinnungsfaktoren = nein **and** SedativaNarkotika = ja **and** NoradrenalinEinmalgabe = nein **then** Klasse verstorben **with** confidence 84.21% **and** frequency 27.54%“

7) „**if** EK = ja **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opioiden **and** Antibiotika = ja **and** Antikoagulation = ja **and** PPSB = nein **and** einzelne Gerinnungsfaktoren = nein **and** SedativaNarkotika = ja **and** Dialyse = nein **and** EnteraleErnaehrung = nein **and** Albumin20(Einfuhr) = nein **and** NoradrenalinEinmalgabe = nein **then** Klasse verstorben **with** confidence 83.33% **and** frequency 26.09%“

12) „**if** DobutaminPerfusor = ja **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opioiden **and** Antihypotonika = nein **and** Antikoagulation = ja **and** PPSB = nein **and** einzelne Gerinnungsfaktoren = nein **and** EnteraleErnaehrung = nein **and** Albumin20(Einfuhr) = nein **and** NoradrenalinEinmalgabe = nein **then** Klasse verstorben **with** confidence 88.24% **and** frequency 24.64%“

6) „if EK = ja **and** FFP = ja **and** DobutaminPerfusor = ja **and** ParenteraleErnaehrung = ja **and** Analgetika = ja **and** Analgetika UK Opioide **and** Antibiotika = ja **and** Antikoagulation = ja **and** Antipyretika = nein **and** PPSB = nein **and** einzelne Gerinnungsfaktoren = nein **and** SedativaNarkotika = ja **and** Beatmungsform = ja **and** Dialyse = nein **and** NoradrenalinEinmalgabe = nein **then** Klasse verstorben **with** confidence 82.35% **and** frequency 24.64%“

Natürlich wären kürzere Regeln mit weniger Items wünschenswert, allerdings sind bei einem so komplexen Problem und der eher geringen Patientenzahl längere Regeln notwendig. Auch scheint es der klinischen Praxis zu entsprechen, dass eine ganze Reihe von Faktoren überprüft wird, um zu entscheiden, ob ein Patient eher kritisch ist. Merken kann sich diese Regeln natürlich niemand, so dass es für eine praktische Verwertbarkeit sinnvoll ist, diese Regeln in einem Expertensystem dem Arzt zur Verfügung zu stellen, das nach Eingabe der Daten entsprechende Regeloptionen anzeigt. Auf jeden Fall positiv zu bewerten sind die guten Konfidenzwerte und die sehr hohen Häufigkeitswerte der Regeln. Der einzige Grund, der gegen die Verwendung eines solchen Systems spricht, ist die Tatsache, dass die Medikament- und Therapie-Items keine Frühwarnung liefern, sondern den Arzt erst nach seiner Medikamentverabreichung bzw. nach seiner Therapie warnen würden. Im Rahmen dieser Arbeit sollen deshalb die Ergebnisse dieses Abschnitts zunächst nur als Erklärung dienen, welche Medikamente bzw. Therapien bei verstorbenen bzw. überlebenden Patienten in Kombination vorkamen. Die Operationen spielten generell eine untergeordnete Rolle.

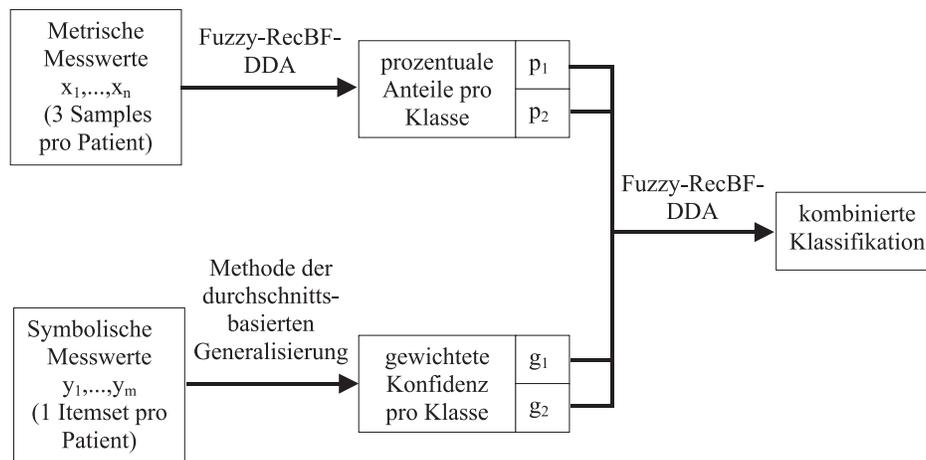
## 7.4 Kombination von Regelmengen

Im Anschluss an die Diskussion des Abschnitts 2.5.3 wird an dieser Stelle ein Versuch zur Kombination von Regelmengen unternommen, der speziell auf die Situation der Klinikdaten zugeschnitten ist.

Prinzipiell gelten die Vereinbarungen aus dem Abschnitt 2.5.5, d.h. die Ausführungen zur Beschaffenheit der metrischen Regelmenge  $R$  und der symbolischen Regelmenge  $S$ .

### **Kombination von SystDiasThrombo3K und Sym35:**

Speziell bei den Klinikdaten muss von folgender Situation ausgegangen werden: Die *metrischen Daten* liegen nach der Datenvorverarbeitung (insbesondere Sampling und fehlende Werte-Ergänzung) als Menge von Tagesmesswert-Vektoren vor. Diese Vektoren sind nach der Datenvorverarbeitung vollständig. Für den folgenden Versuch wird die repräsentative Regelmenge des Datensatzes SystDiasThrombo3K verwendet (das ist die Regelmenge des ersten Versuchs, vgl. Tab. 7.23), die die besten Ergebnisse lieferte. Sie enthält 17 Regeln (6 für „überlebend“, 9 für „verstorben“).



**Abbildung 7.21:** Schema des kombinierten Systems zur gemeinsamen Klassifikation von metrischen und symbolischen Daten für die Datensätze SystDiasThrombo3K und Sym35.

Die Regelmenge der *symbolischen Daten* ist hier anders aufgebaut. Da pro Tag nur einzelne oder wenige symbolische Variablen aus einer sehr großen Menge (alle möglichen Operationen aus dem OP301-Code, ausgewählte Diagnosen und Medikamente) von binären symbolischen Variablen bei einem Patienten auftreten, wurden die symbolischen Daten (sowohl im Kap. 6 als auch im Kap. 7) über die Zeit zu einem Itemset pro Patient zusammengefasst, so dass ein Patient eine größere Anzahl von zutreffenden JA-Items hatte. Das führte zu der repräsentativen Regelmenge  $S$  (die erste Regelmenge der 3 Versuche), die aus 35 Regeln besteht (15 für „überlebend“, 20 für „verstorben“), vgl. den Abschnitt 7.3.2. Der dort verwendete Datensatz, der durch eine Menge von Itemsets repräsentiert wird, sei im Folgenden als „Sym35“ bezeichnet.

Für die beiden beschriebenen Regelmengen  $R$  und  $S$  ist die Tupeleigenschaft aus der Vereinbarung 2.5.5 verletzt. Die metrischen Daten sind Messwerte pro Tag pro Patient; die symbolischen Items sind Werte, die pro Patient gelten, aber nicht für einen bestimmten Tag. Man kann allerdings, um überhaupt einen Versuch durchführen zu können, das symbolische Itemset für einen Patienten jedem der drei metrischen SystDiasThrombo3K-Messwert-Vektoren des Patienten zuordnen, so dass man dadurch die Tupeleigenschaft künstlich herstellt.

Dann ist eine Klassifikation mit den Ausgaben, d.h. den unscharfen Klassenzugehörigkeiten, unter Verwendung eines weiteren Fuzzy-RecBF-DDA-Netzes (vgl. Abb. 7.21) möglich zur Erzeugung von Klassifikationsregeln der Form: „**if**  $y_1^{(m)}$  hat prozentualen Anteil **in**  $[q_1, q_2]$  **and**  $y_2^{(m)}$  hat prozentualen Anteil **in**  $[q_3, q_4]$  **and**  $y_1^{(s)}$  hat gewichtete Konfidenz **in**  $[h_1, h_2]$  **and**  $y_2^{(s)}$  hat gewichtete Konfidenz **in**  $[h_3, h_4]$  **then**

	Mittelwert	Standardabw.	Minimum	Maximum
Training korrekt	86.60	0.97	85.86	88.24
Training nicht klassifiziert	0.30	0.68	0.00	1.52
Test korrekt	82.47	2.24	79.00	85.28
Test nicht klassifiziert	0.69	1.55	0.00	3.47

**Tabelle 7.30:** SystDiasThrombo3K/Sym35-Daten: Korrekte Klassifikationen auf den Trainings- und Testdaten, nicht klassifizierte Trainings- bzw. Testdaten. Angaben in Prozent.

Klassenzugehörigkeit  $K$  **with** Gütemaße (Häufigkeit und Konfidenz)“, vgl. den Abschnitt 2.5.3.

Zu diesem Versuchsaufbau sollte etwas zum Training und Testen gesagt werden: Die einzelnen Systeme wurden mit jeweils 50% der Daten trainiert und mit den anderen 50% der Daten getestet, so dass bereits alle Daten verwendet wurden. Da aber jetzt noch das zusätzliche Netz zur Kombination getestet werden muss, kann grundsätzlich der metrische und der symbolische Regelgenerator mit einem Drittel der Daten trainiert werden und mit einem Drittel getestet werden. Dieses Drittel kann zum Trainieren des Kombinationsnetzes verwendet werden. Das letzte Drittel wird dann zum Testen des Kombinationssystems verwendet. Dieses Vorgehen ist sinnvoll, um unabhängig vom Training die Leistung des Systems zu beurteilen.

Von diesem i. Allg. sinnvollen Vorgehen wird an dieser Stelle aber abgewichen. Zum einen würde eine Dreiteilung der Daten in unserem Fall zu kleinen Trainings- und Testmengen führen, insbesondere bei den symbolischen Itemsets, und zum anderen möchten wir die Ergebnisse der bereits trainierten und getesteten repräsentativen Regelmengen zur Kombination verwenden. Wir werden deshalb für alle Daten die Ausgaben der Einzelsysteme berechnen und anschließend zum Trainieren des zweiten Fuzzy-RecBF-DDA-Netzes 50% dieser Ausgabedaten verwenden und die anderen 50% zum Testen; wiederum soll die Trainings- und Testmenge bez. der Patienten disjunkt sein. Das mit einem Teil der Daten dann die Einzelsysteme und das Kombinationssystem trainiert wurden, wird das Ergebnis vermutlich kaum beeinflussen, da die Klassifikationsergebnisse der jeweiligen Einzelsysteme sich auf den Trainings- und Testdaten um weniger als 1% unterschieden haben.

**Versuchseinstellungen:** vorgegebene Klassifikationsleistung  $\text{errorcriteria} = 78$ , Ergebnisse siehe Tabellen 7.30 und 7.31 (Klassifikation), Tabelle 7.32 (Regelgenerierung). Die verwendeten Testdaten kamen in den 5 Testdurchläufen im Mittel zu 49.30% aus Klasse  $\ddot{u}$  und zu 50.70% aus Klasse  $v$ . Die Epochenzahl betrug im Mittel 3.0 mit einer Standardabweichung von 0 (Minimum 3, Maximum 3).

	Klasse ü	Klasse v
Klasse ü (Training)	42.30	4.69
Klasse v (Training)	8.40	44.30
Klasse ü (Test)	37.48	5.02
Klasse v (Test)	11.82	44.99

**Tabelle 7.31:** SystDiasThrombo3K/Sym35-Daten: Fehler auf den Trainings- und Testdaten (Spalte: Klassenzugehörigkeit, Zeile: Ergebnis der Klassifikation). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü	8.2	1.5	6	10
Klasse v	6.2	1.1	5	8

**Tabelle 7.32:** SystDiasThrombo3K/Sym35-Daten: Anzahl der erzeugten Regeln pro Klasse (Mittelwert, Standardabweichung, Minimum, Maximum).

Um das Klassifikationsergebnis im Hinblick auf die Veränderung der Klassifikationsschwelle besser einschätzen zu können, werden in der Abb. B.9 alle Klassifikationsergebnisse der 5 Versuche aufgetragen, wobei jeweils 10 Punkte eingetragen wurden, also insgesamt 50 Klassifikationen durchgeführt wurden. Für die SystDiasThrombo3K/Sym35-Daten erhält man den sehr guten Wert 0.921 (Min.: 0.898, Max.: 0.940). Das ist ein etwas besseres Ergebnis als das bez. des SystDiasThrombo3K-Datensatzes, vgl. S. 225. Dort betrug der mittlere ROC-Flächeninhalt 0.895.

Zwei interessante Beispielregeln (Regeln 1 und 17) aus dem repräsentativen vierten der 5 Versuche (Niveau 0.0 und 1.0), die die weiter o.g. spezielle Form haben und die eine sehr gute Häufigkeit und Konfidenz besitzen und auch Daten von vielen Testpatienten abdecken, seien an dieser Stelle genannt:

1) support class **survived** with test frequency **0.361** and test confidence **0.877** from **27** different patients if  
**var** *Proz. Anteil (metrisch) Kl. ü*  $\geq 16.33$  and  
**var** *Gew. Konfidenz (symbolisch) Kl. ü*  $\geq 0.71$  and  
**var** *Gew. Konfidenz (symbolisch) Kl. v*  $\leq 0.26$   
core class **survived** with test corefrequency **0.208** and test coreconfidence **0.976** from **18** different patients if  
**var** *Proz. Anteil (metrisch) Kl. ü* in (60.46,100.00) and  
**var** *Proz. Anteil (metrisch) Kl. v*  $\leq 57.53$  and  
**var** *Gew. Konfidenz (symbolisch) Kl. ü* in (0.77,0.85) and  
**var** *Gew. Konfidenz (symbolisch) Kl. v* in (0.15,0.23)

17) support class **deceased** with test frequency **0.089** and test confidence **0.833** from **11** different patients if

**var** *Proz. Anteil (metrisch) Kl. v*  $\leq 47.79$  and

**var** *Gew. Konfidenz (symbolisch) Kl.  $\ddot{u}$*   $\leq 0.85$  and

**var** *Gew. Konfidenz (symbolisch) Kl. v* in  $(0.50, 0.80)$

core class **deceased** with test corefrequency **0.015** and test coreconfidence **1.000** from **2** different patients if

**var** *Proz. Anteil (metrisch) Kl.  $\ddot{u}$*  in  $(67.87, 84.40)$  and

**var** *Proz. Anteil (metrisch) Kl. v* in  $(15.60, 25.52)$  and

**var** *Gew. Konfidenz (symbolisch) Kl.  $\ddot{u}$*  in  $(0.20, 0.35)$  and

**var** *Gew. Konfidenz (symbolisch) Kl. v* in  $(0.65, 0.80)$

Bei der Betrachtung aller Regeln erkennt man, dass durchaus metrische *und* symbolische Variablen ihren Teil zur Klassifikation beitragen. Ist der prozentuale Anteil bzw. die gewichtete Konfidenz für eine Klasse hoch, so ist er bzw. sie für die andere Klasse eher niedrig; es treten aber Überlappungen auf.

**Fazit:** Die symbolischen Daten ließen sich besser pro Patient, d.h. ohne Zeitstempel, analysieren, da Operationen o.ä. eher selten auftreten. Messwerte wie der Blutdruck werden ständig gemessen und somit sind deutlich mehr metrische Samples pro Patient vorhanden als symbolische. Das führt zur Verletzung der Tupeleigenschaft bei der Kombination von Regelmengen, so dass ein individueller Ansatz zur Kombination der Regelmengen zur Klassifikation mit zusätzlichen Klassifikationsregeln gewählt wurde. Für metrische und symbolische Daten wurden (bereits in den vorhergehenden Kapiteln) *getrennt* Regeln ausgegeben. In diesem Abschnitt wurde dann untersucht, ob die Klassifikationsleistung durch die kombinierten Regelmengen steigt.

Bei den durchgeführten Analysen (insbesondere der Klassifikationsleistung) wurde festgestellt, dass die metrischen Daten allein bereits gut zur Klassifikation geeignet sind. Die bis zu 35 zusätzlichen symbolischen Werte (die ein Arzt zusätzlich angeben müsste) führten nur zu einer Verbesserung des ROC-Flächeninhalts von  $0.921 - 0.895 = 0.026$ . Vergleicht man die Ergebnisse der Tabellen 7.30 und 7.20, so ist die Standardabweichung in der Tab. 7.30 etwas niedriger; es werden aber mehr (Güte-)daten als nicht klassifizierbar eingestuft. Auf jeden Fall wird die Klassifikationsleistung durch die metrischen Daten (ROC-Fläche: 0.895) im Vergleich zu den symbolischen (ROC-Fläche: 0.761) „dominiert“ (ROC-Fläche der Kombination: 0.921).

Insgesamt ist es sinnvoll, aus den metrischen Daten eine Alarmgebung zu generieren ohne die symbolischen Daten zu berücksichtigen (Abschnitt 7.5). Der Arzt muss dann nur die körpereigenen Variablen eingeben und kann ganz auf die Eingabe von symbolischen Daten (Operationen, Therapien, Medikamente) verzichten.

Ein Vorschlag, der zu einer Berücksichtigung der Operationen innerhalb der metrischen Regelgenerierung führen kann, lautet wie folgt: Pro Operation könnte man einen *Schweregrad* für Operationen definieren. Individuell nimmt für jede Operation die Belastung für den Patienten durch die Operationen im Laufe der Zeit zu bzw. ab. So könnte man die Einzelbelastungen durch die Operationen für den Patienten pro Zeiteinheit (z.B. pro Tag) zusammenfassen und als metrischen Wert definieren. Dann kann man die Variable „Operationsbelastung“ zusammen mit den anderen metrischen Variablen betrachten. Vorteilhaft wäre die damit verbundene Dimensionsreduktion von einer Vielzahl von (kaum aussagekräftigen) Operationen auf eine, vielleicht aussagekräftigere, Variable. Analog könnte man eine „Therapie und/oder Medikamentenschwere“ definieren.

## 7.5 Aufbau eines Alarmsystem auf der Basis der Fuzzy-RecBF-DDA-Regelmengen

Auf der Grundlage der Ergebnisse, die mit den metrischen Klinikdaten im Abschnitt 7.2 erzielt wurden, soll nun anhand der repräsentativen Regelmengen die Leistungsfähigkeit darauf aufbauender Alarmsysteme bewertet werden.

Eine Kombination mit den Regelmengen der symbolischen Daten aus dem Abschnitt 7.3 wäre prinzipiell möglich. Es gibt aber entscheidende Gründe, darauf zu verzichten:

a) In einem Realweltssystem müsste zusätzlich eine beträchtliche Anzahl von symbolischen Daten eingegeben werden; Ärzte auf der Intensivstation wären dazu nicht bereit.

b) Die symbolischen Regeln basieren auf Medikamentgaben, Therapien, Diagnosen, also alles Variablen, deren Wahrheitswerte vom Arzt bestimmt werden. D.h. diese Regeln liefern dem Arzt Informationen über die Interaktion mit dem Patienten. Wünschenswert für ein Alarmsystem ist es, dass körpereigene Variablen (Messwerte) den Alarm auslösen. Eine symbolische Alarmregel der Art „**if** Arzt verabreicht Patient eine hohe Dosis Katecholamine **then** Patient ist sehr kritisch“ ist als Aussage einer retrospektiven Auswertung interessant; auf der Intensivstation bekommt der Patient ja gerade die Katecholamine, weil er sehr kritisch ist.

c) Im letzten Abschnitt wurde gezeigt, dass eine Kombination die Klassifikationsleistung nur geringfügig verbessert.

Folgerichtig erscheint es sinnvoll, ein Alarmsystem aus den metrischen Werten aufzubauen. Die symbolische Regelgenerierung als eigenständige Analyse lieferte einige interessante Ergebnisse, insbesondere über den Gebrauch von Medikamenten

und Therapien; die metrischen (Messwert-)Regeln eignen sich besser für ein Frühwarnsystem. Also wende ich mich wieder den metrischen Klinikdaten zu, und zwar den Datensätzen Häufig16K, SOFA1K, SystThrombo2K, Wichtig6K und SystDiastThrombo3K (zum Zeitraum L3). Überprüft werden soll, inwieweit sich sinnvolle Alarme für die Patientenzeitreihen erzeugen lassen.

Als Erstes ist zu klären, wie Alarme generiert werden können und in welcher Form. Anschließend ist zu klären, wie die Güte der Alarme überprüft werden kann. Zum Schluss werden die Ergebnisse der Alarmsysteme verglichen, die aus den fünf Datensätzen erzeugt wurden.

### 7.5.1 Erzeugung der Alarme

Nach Voruntersuchungen hat sich herausgestellt, dass zwei unterschiedliche Alarmstufen sinnvoll sind (mehr verwirren den Arzt und sind datentechnisch nicht zu begründen, weniger sind recht ungenau). Wir gehen also von zwei Alarmstufen 1 und 2 aus mit der Bedeutung  $1 = \textit{kritisch}$  und  $2 = \textit{sehr kritisch}$ . Die Definitionen lassen sich z.B. anhand der Abb. 7.22 wie folgt begründen: Alarme der Stufe 2 charakterisieren Patientenzustände, die besonders häufig bei Verstorbenen auftreten, aber sehr selten (insbesondere am Ende der Liegedauer) bei Überlebenden. Alarme der Stufe 1 werden öfter auch für Überlebende generiert. Die Alarme für Überlebende sind aber i. Allg. nicht als Fehlalarme zu interpretieren, da sich auch die Überlebenden zeitweise in kritischen Zuständen befinden können.

Der modifizierte Fuzzy-RecBF-DDA-Regelgenerator liefert bei zwei Klassen in der Anwendungsphase für einen Eingabevektor zwei Aktivierungen, eine für die Klasse  $\bar{u}$  (überlebt) und eine für die Klasse  $v$  (verstorben), die mit  $y_1$  und  $y_2$  bezeichnet seien. Da die Aktivierungen sehr unterschiedlich hohe Absolutwerte annehmen können, sind die Werte  $y_1, y_2$  so noch nicht geeignet als Schwellwerte, bei deren Überschreitung ein Alarm ausgelöst wird. Berechnet wird nun der prozentuale Anteil  $p_i$ ,  $i = 1, 2$ , an der Summe der Aktivierung, also  $p_i := y_i/s_y \cdot 100$  mit  $s_y := y_1 + y_2$  (für  $s_y \neq 0$ ). Ist  $s_y = 0$ , so wird  $p_i = 50$  gesetzt für beide Klassen.

Für zwei Schwellen  $k_1 < k_2 \in [0, 100]$  wird vereinbart: Ein sehr kritischer Alarm der Stufe 2 wird für eine Eingabe  $x$  generiert, falls  $p_2 - p_1 \geq k_2$  gilt. Ist  $k_2 > p_2 - p_1 \geq k_1$ , so wird ein Alarm der Stufe 1 generiert. Gilt  $p_2 - p_1 < k_1$ , so wird der Alarmstatus 0 (unkritisch bzw. unbekannt) generiert.

Für jeden Patienten  $P$  mit der zugehörigen Patientenzeitreihe  $Z$  entsteht nun eine *Alarmhistorie*, d.h. jedem Zeitpunkt der Zeitreihe  $Z$  wird ein *Alarmstatus* zugeordnet; der Alarmstatus kann sein:  $0 = \textit{unkritisch bzw. unbekannt}$ ,  $1 = \textit{kritisch}$ ,  $2 = \textit{sehr kritisch}$ , aber auch  $-9999 = \textit{nicht berechenbar}$  aufgrund zu vieler fehlender Werte.

Der Alarmstatus ist berechenbar, falls mindestens pro Zeitpunkt die Anzahl der Variablen aus der Tab. 7.5 vorliegt. Da ein 24h-Sampling verwendet wurde, kann hier also pro Tag ein Alarmstatus generiert werden.

**Güte der Alarme:** Entscheidend für die Güte der Alarmhistorien ist deren medizinische Plausibilität und eine gewisse Stabilität (mehr Alarme für Verstorbene, weniger für Überlebende). Die medizinische Plausibilität ist schwer zu überprüfen und kann im Rahmen dieser Arbeit nicht erfolgen, da Ärzte sämtliche Patientenakten auf kritische Zustände durchsuchen müssten und selbst diese aufwendige Analyse wäre subjektiv, da ja die Alarme auf der Basis der Daten erstellt wurden und nicht unbedingt mit dem medizinischen Wissen übereinstimmen müssen. Die Alarmhistorien wurden von mir nach folgenden Kriterien auf ihre Plausibilität hin untersucht:

- möglichst viele Alarme zum Ende hin bei verstorbenen Patienten, auch möglichst der Stufe 2,
- möglichst wenige Alarme zum Ende hin bei überlebenden Patienten,
- insgesamt möglichst öfter der Alarmstatus 1 oder 2 als 0 bei Zeitreihen von Verstorbenen,
- insgesamt möglichst öfter der Alarmstatus 0 als 1 oder gar 2 bei Zeitreihen von Überlebenden.

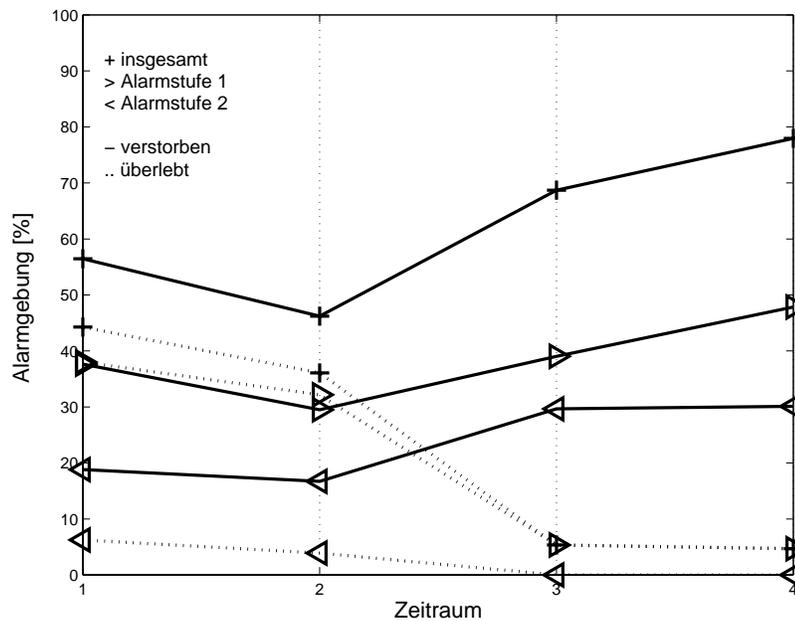
Selbstverständlich ist eine Zeitreihe eines Überlebenden plausibel, wenn der Patient am ersten oder zweiten Tag Alarme erhalten hat, aber danach unkritisch war oder eines Verstorbenen, der erst am Ende Alarme erhält, am Anfang aber nicht unbedingt. Mit diesen Kriterien habe ich subjektiv die Zeitreihen bewertet: *plausibel* oder *unplausibel*. Desweiteren werden in den folgenden Abbildungen 7.22 bis 7.25 jeweils für die ersten 3 Tage, für die erste Hälfte der Liegedauer, für die zweite Hälfte der Liegedauer und für die letzten 3 Tage folgende Kennzahlen eingetragen: Der prozentuale Anteil der Warnungen der Stufe 1 bzw. 2 bzw. 1 und 2 gemeinsam an allen berechneten Alarmstatuswerten (0,1,2 ohne -9999), getrennt für Überlebende und Verstorbene.

Zur Überprüfung der Alarme werden die Patienten aus der Testmenge der repräsentativen Regelmenge (siehe Tab. 7.23) verwendet, da dies die Regelmenge der Wahl für eine praktische Anwendung wäre.

## 7.5.2 Alarmhistorien auf den Klinikdatensätzen

Für folgende Datensätze, die aufgrund ihrer Klassifikationsergebnisse interessant erscheinen, werden die generierten Alarme untersucht:

- Häufig16K,



**Abbildung 7.22:** Häufig16K-Daten: Alarmkennzahlen. Kritische Alarme der Stufen 1 und 2 sowie Alarme insgesamt für verstorbene und überlebende Patienten. Mittelwerte über die Patienten der Testdaten. Zeitpunkte: 1 = erste 3 Tage, 2 = erste Hälfte der Liegedauer, 3 = zweite Hälfte der Liegedauer, 4 = letzte 3 Tage.

- SOFA1K,
- SystThrombo2K,
- Wichtig6K und
- SystDiasThrombo3K.

#### Alarmhistorien für den Datensatz Häufig16K:

Die o.g. Alarmkennzahlen für den Häufig16K-Datensatz findet man in der Abb. 7.22 und numerisch in der Tab. C.31. Die (subjektive) Plausibilität der Zeitreihen findet man in der Tab. 7.33. 63 Testpatienten wurden überprüft. Die Schwellen lauteten:  $p_1 = 15$  und  $p_2 = 85$ .

Unplausible Patienten bei den Überlebenden gab es fast nie. Das ist gut. Das System warnt fast nie, wenn der Patient unkritisch ist.

Bei den unplausiblen verstorbenen Patienten war bei 6 von 8 Verstorbenen am letzten Tag ein fehlender Status (-9999) vorhanden, d.h. es ist zu vermuten, dass hier wie aber auch bei den anderen beiden unplausiblen Verstorbenen am Ende des Intensivaufenthalts ein plötzlicher Tod stand, dessen Dynamik deutlich schneller als

	plausibel	unplausibel	plausibel [%]	unplausibel [%]
überlebt	31	1	96.6	3.4
verstorben	23	8	74.2	25.8
insgesamt	54	9	85.7	14.3

**Tabelle 7.33:** Häufig16K-Daten: Plausible Alarme für überlebende, verstorbene, alle Patienten. Angaben in Prozent.

24h ablief, so dass z.B. keine Zeit mehr für Messungen und Dokumentation auf der Intensivstation vorhanden war, so dass das Alarmsystem diese Kurzzeitdynamik nicht erfassen kann. Für diese Patienten kann ein Alarmsystem nur mit Online-monitoring-Daten aufgebaut werden; die retrospektive Analyse von Patientenakten reicht hier nicht aus. Möglich wäre aber auch, dass von Seiten der Ärzte bewusst keine Behandlung mehr durchgeführt wurde und damit keine Messwerte des Patienten vorhanden sind (Therapia Minima).

#### Typische Phänomene:

Anhand der Zeitreihen für die Liegedauer der Patienten werden einige typische Alarmhistorien vorgestellt:

**Patient A:** 8 Tage Liegedauer, überlebt

0 0 0 0 0 0 0 -9999 (Jeder Wert entspricht einem Alarmstatus für einen Tag.)

plausibel: Patient ist unkritisch während seines Intensivaufenthalts (im Sinne, dass er nicht versterben wird).

**Patient B:** 11 Tage Liegedauer, überlebt

1 1 0 0 0 0 0 0 0 -9999

plausibel: auch wenn er am Anfang 2 Tage lang kritisch war.

**Patient C:** 4 Tage Liegedauer, verstorben

1 2 2 2

plausibel: während der Liegedauer sehr kritisch.

**Patient D:** 9 Tage Liegedauer, verstorben

1 0 1 2 1 1 1 2 2

plausibel: wird während der Liegedauer sehr kritisch.

**Patient E:** 19 Tage Liegedauer, verstorben

0 0 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 -9999

plausibel: fast während der ganzen Liegedauer kritisch.

**Patient F:** 14 Tage Liegedauer, überlebt

0 1 1 1 1 0 0 1 1 0 1 1 1 -9999

unplausibel: einziger unplausibler überlebender Patient; evtl. wäre die -9999 eine 0 gewesen, d.h. dieser Patient ist nicht völlig unplausibel (nicht auszuschließen ist natürlich, dass der Patient in einem noch recht kritischen Zustand bereits die Intensivstation verlassen hat und auf eine sog. Normalstation verlegt wurde).

**Patient G:** 13 Tage Liegedauer, verstorben

0 0 1 0 0 0 0 0 -9999 0 0 0 1

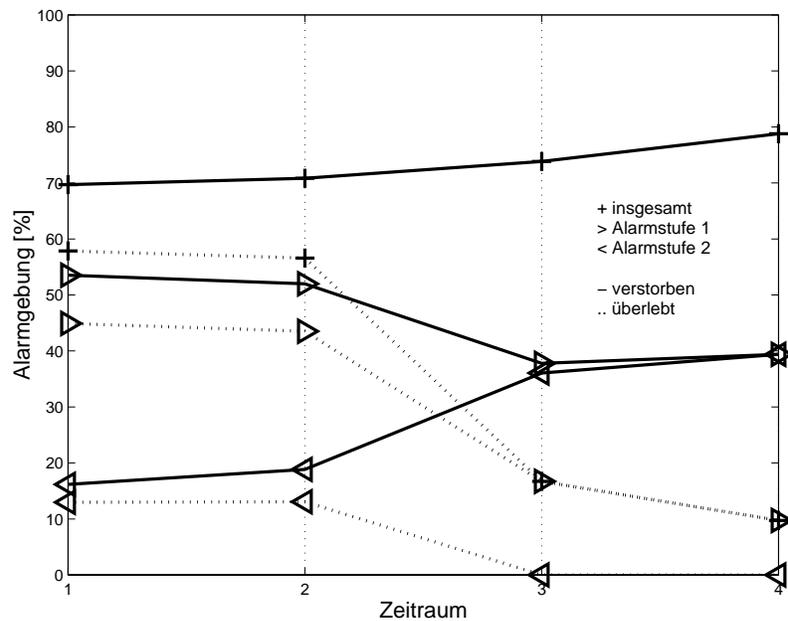
unplausibel: der Patient scheint recht unkritisch zu sein während seiner Liegedauer, auch wenn am Ende noch eine Warnung erfolgt.

**Patient H:** 13 Tage Liegedauer, verstorben

0 0 0 0 0 0 0 0 1 0 1 1 -9999

unplausibel: am Ende erfolgen zwar Warnungen, aber keine der Stufe 2, allerdings hat der letzte Tag den Status -9999, so dass nicht klar ist, ob das System gewarnt hätte.

Insgesamt sind die meisten Alarmhistorien plausibel, einige weniger plausibel (hier konservativ als unplausibel gekennzeichnet). Die Gründe dafür wurden bereits im Verlauf des Textes genannt.



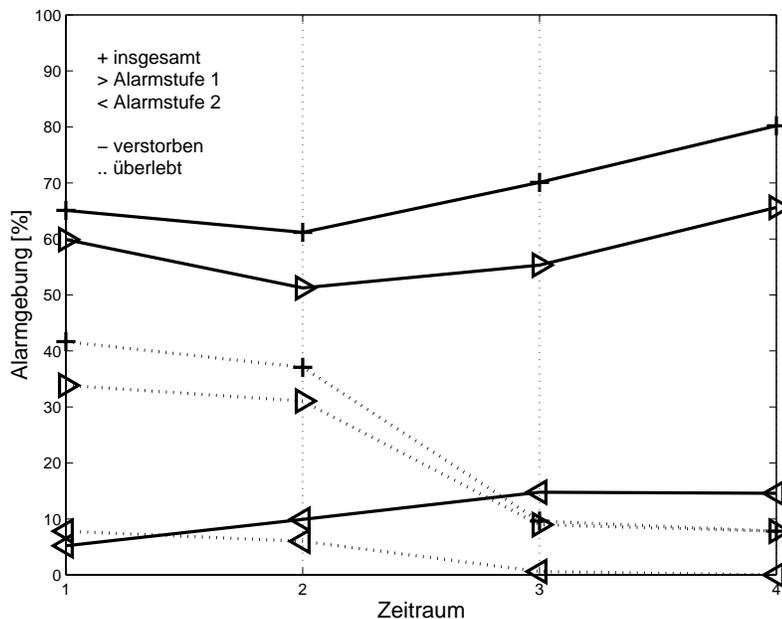
**Abbildung 7.23:** SOFA1K-Daten: Alarmkennzahlen. Kritische Alarme der Stufen 1 und 2 sowie Alarme insgesamt für verstorbene und überlebende Patienten. Mittelwerte über die Patienten der Testdaten. Zeitpunkte: 1 = erste 3 Tage, 2 = erste Hälfte der Liegedauer, 3 = zweite Hälfte der Liegedauer, 4 = letzte 3 Tage.

### Alarmhistorien für den Datensatz SOFA1K:

Die Alarmkennzahlen für den SOFA1K-Datensatz findet man in der Abb. 7.23 und numerisch in der Tab. C.32. Die (subjektive) Plausibilität der Zeitreihen findet man in der Tab. 7.34. 69 Testpatienten wurden überprüft. Die Schwellen lauteten:  $p_1 = 40$  und  $p_2 = 95$ .

	plausibel	unplausibel	plausibel [%]	unplausibel [%]
überlebt	33	3	91.7	8.3
verstorben	24	9	72.7	17.3
insgesamt	57	12	82.6	17.4

**Tabelle 7.34:** SOFA1K-Daten: Plausible Alarme für überlebende, verstorbene, alle Patienten.



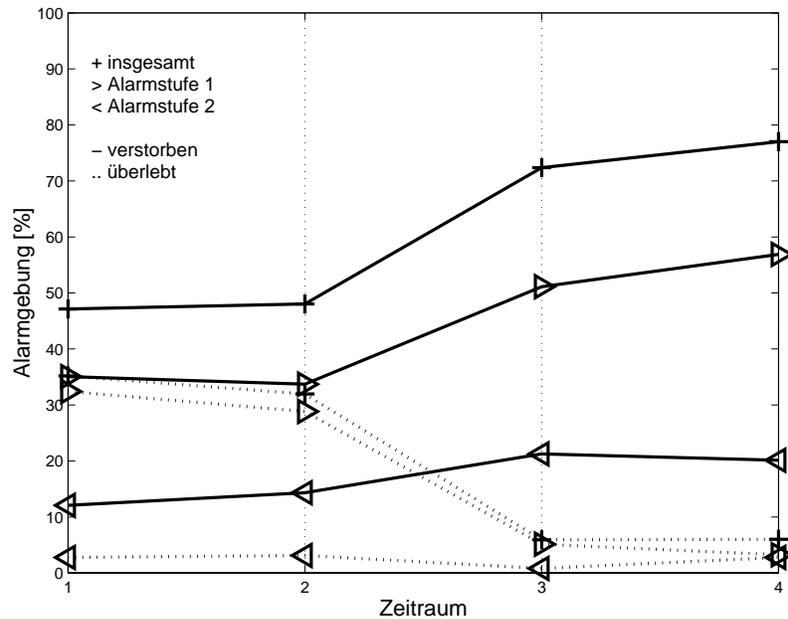
**Abbildung 7.24:** SystThrombo2K-Daten: Alarmkennzahlen. Kritische Alarmer der Stufen 1 und 2 sowie Alarmer insgesamt für verstorbene und überlebende Patienten. Mittelwerte über die Patienten der Testdaten. Zeitpunkte: 1 = erste 3 Tage, 2 = erste Hälfte der Liegedauer, 3 = zweite Hälfte der Liegedauer, 4 = letzte 3 Tage.

### Alarmhistorien für den Datensatz SystThrombo2K:

Die Alarmkennzahlen für den SystThrombo2K-Datensatz findet man in der Abb. 7.24 und numerisch in der Tab. C.33. Die (subjektive) Plausibilität der Zeitreihen findet man in der Tab. 7.35. 64 Testpatienten wurden überprüft. Die Schwellen lauteten:  $p_1 = 40$  und  $p_2 = 90$ .

	plausibel	unplausibel	plausibel [%]	unplausibel [%]
überlebt	30	2	93.8	6.2
verstorben	23	9	71.9	28.1
insgesamt	53	11	82.8	17.2

**Tabelle 7.35:** SystThrombo2K-Daten: Plausible Alarmer für überlebende, verstorbene, alle Patienten.



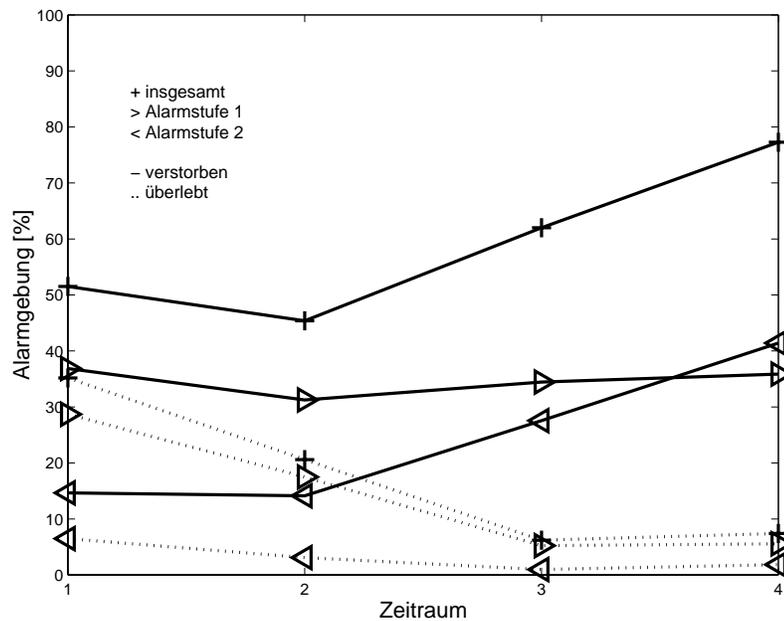
**Abbildung 7.25:** Wichtig6K-Daten: Alarmkennzahlen. Kritische Alarme der Stufen 1 und 2 sowie Alarme insgesamt für verstorbene und überlebende Patienten. Mittelwerte über die Patienten der Testdaten. Zeitpunkte: 1 = erste 3 Tage, 2 = erste Hälfte der Liegedauer, 3 = zweite Hälfte der Liegedauer, 4 = letzte 3 Tage.

### Alarmhistorien für den Datensatz Wichtig6K:

Die Alarmkennzahlen für den Wichtig6K-Datensatz findet man in der Abb. 7.25 und numerisch in der Tab. C.34. Die (subjektive) Plausibilität der Zeitreihen findet man in der Tab. 7.36. 65 Testpatienten wurden überprüft. Die Schwellen lauteten:  $p_1 = 20$  und  $p_2 = 70$ .

	plausibel	unplausibel	plausibel [%]	unplausibel [%]
überlebt	34	2	94.4	5.6
verstorben	22	7	75.9	24.1
insgesamt	56	65	86.2	13.8

**Tabelle 7.36:** Wichtig6K-Daten: Plausible Alarme für überlebende, verstorbene, alle Patienten.



**Abbildung 7.26:** SystDiasThrombo3K-Daten: Alarmkennzahlen. Kritische Alarme der Stufen 1 und 2 sowie Alarme insgesamt für verstorbene und überlebende Patienten. Mittelwerte über die Patienten der Testdaten. Zeitpunkte: 1 = erste 3 Tage, 2 = erste Hälfte der Liegedauer, 3 = zweite Hälfte der Liegedauer, 4 = letzte 3 Tage.

### Alarmhistorien für den Datensatz SystDiasThrombo3K:

Die Alarmkennzahlen für den SystDiasThrombo3K-Datensatz findet man in der Abb. 7.26 und numerisch in der Tab. C.35. Die (subjektive) Plausibilität der Zeitreihen findet man in der Tab. 7.37. 69 Testpatienten wurden überprüft. Die Schwellen lauteten:  $p_1 = 8$  und  $p_2 = 60$ .

	plausibel	unplausibel	plausibel [%]	unplausibel [%]
überlebt	32	4	88.9	11.1
verstorben	28	5	84.9	15.1
insgesamt	60	9	87.0	13.0

**Tabelle 7.37:** SystDiasThrombo3K-Daten: Plausible Alarme für überlebende, verstorbene, alle Patienten.

### 7.5.3 Zusammenfassende Diskussion über die Alarmsysteme

In den Analysen wurde die Plausibilitätsleistung der Alarmhistorien der Datensätze bewertet. Vergleicht man diese Plausibilitätsleistung, so erhält man folgende Platzierung der Alarmsysteme für die betrachteten Datensätze:

1. SystDiasThrombo3K: 87.0% (jeweils „Datensatz: Plausibilität“),
2. Wichtig6K: 86.2%,
3. Häufig16K: 85.7%,
4. SystThrombo2K: 82.8% und
5. SOFA1K: 82.6%.

Man sieht, dass mit weniger als den 10 Variablen des SOFA-Scores plausible Alarmhistorien erzeugt werden können (SystDiasThrombo3K, SystThrombo2K und Wichtig6K). Die Plausibilität auf den fünf Datensätzen ist recht ähnlich. Da die Plausibilität subjektiv beurteilt wurde, die Warnungen von Schwellen abhängig sind und jeweils andere Testpatienten in die Bewertung eingegangen sind, kann nicht unbedingt von signifikanten Unterschieden gesprochen werden.

Dennoch hat der Datensatz SystDiasThrombo3K (Abb. 7.26) die besten Eigenschaften für die Alarmgebung: Dort können am Ende sogar mehr Warnungen der Stufe 2 ausgesprochen werden als der Stufe 1 ohne den Fehllalarm zu erhöhen. Das Ergebnis ist in diesem Sinn stabil und plausibel; 3 Variablen sind weniger als 10 Variablen, die für die Berechnung der SOFA-Variable benötigt werden. Die drei Variablen sind leicht auf der Intensivstation zu messen. Insofern spricht alles für die Verwendung des Datensatzes SystDiasThrombo3K für die Konstruktion eines Alarmsystems.

Bis zu diesem Ergebnis war es ein langer Weg. Langjähriges Expertenwissen durch gleichwertige oder gar bessere technische Systeme zu ersetzen, ist nicht einfach. In diesem Fall ist es gelungen. Für diesen Zweck musste das Fuzzy-RecBF-DDA-Verfahren für die Praxistauglichkeit implementiert, evaluiert und verbessert werden. Zahlreiche, oft fälschlicherweise als einfach bezeichnete Vorverarbeitungen mussten ausprobiert werden. Neu entwickelte Bewertungen wie die Wichtigkeit von Variablen und die Ähnlichkeit zur Modellwahl wurden mit zur Analyse herangezogen. Zusätzlich wurde die Verwendbarkeit der symbolischen Variablen überprüft, wozu ein neuer Ansatz zur Generierung von Generalisierungsregeln erprobt wurde.

Zu guter Letzt kann mit Sicherheit behauptet werden, dass die Expertenleistung anhand der vorhandenen Daten in guter Qualität nachgebildet wurde. Ob der Einsatz des konstruierten Alarmsystems in der intensivmedizinischen Praxis zu einem medizinisch sichtbaren Erfolg führen wird, kann nur anhand einer medizinischen Studie geklärt werden.

# Kapitel 8

## Zusammenfassung und Ausblick

In der vorliegenden Arbeit wurde ein klinisches *Alarmsystem* für septische Schock-Patienten aufgebaut. Zweckmäßigerweise wurden hierfür metrische, körpereigene Variablen verwendet, da Analysen belegt haben, dass die metrischen Daten besser zur Alarmgenerierung geeignet sind als die symbolischen Daten. Für das Training des adaptiven Neuro-Fuzzy-Systems wurden die Daten der letzten Tage des Intensivaufenthalts verwendet, da in diesem Zeitraum, im Gegensatz zu den ersten Tagen, eine gute Klassifikationsperformanz erreicht wurde. Die daraus resultierenden Alarmhistorien liefern zuverlässige Hinweise für den Intensivmediziner auf besonders kritische Patienten. Durch diese Arbeit wird es möglich werden, den medizinischen SOFA-Score, der aus 10 Variablen zusammengesetzt ist, durch die einfachere Kombination „Systolischer Blutdruck / Diastolischer Blutdruck / Thrombozyten“ zu ersetzen mit einer mindestens genauso guten Performanz. Durch die Hinzunahme weiterer Variablen ist es möglich, die Performanz des SOFA-Scores zu überbieten, wobei der SOFA-Score bereits die beste Klassifikationsperformanz unter den getesteten Scores erreichte. Die erzeugten Regeln konnten die Klassifikationsentscheidung sinnvoll untermauern. Im Gegensatz zur automatischen Regelgenerierung war es Ärzten nicht möglich, ähnlich sinnvolle formale Regeln zu formulieren.

Für das Training des Alarmsystem standen nur 138 Patienten zur Verfügung (Der SOFA-Score wurde in [V<sup>+</sup>96] anhand 1643 Patienten evaluiert). Daher ist es sinnvoll, das System erneut anhand einer größeren Patientenzahl zu evaluieren, um die Ergebnisse noch besser abzusichern. Die Praxistauglichkeit des Systems kann allerdings ausschließlich in einer medizinischen Studie oder Multizentrenstudie überprüft werden. Solch eine Multizentrenstudie ist im Rahmen des MEDAN-Projekts geplant.

Zusätzlich zu den durchgeführten Experimenten könnte für die Analyse des septischen Schocks eine vertiefte Analyse der Zeitreihendynamik sinnvoll sein; diese Möglichkeit wie auch die Möglichkeit einer deutlich früheren Prognose wird aber stark durch die Häufigkeit und die Datenqualität der Messungen eingeschränkt. Eine Verbesserung wäre die Verwendung von Online monitoring-Daten, evtl. mit prospektiven Messungen, anstelle der retrospektiv analysierten Patientenakten.

Insbesondere sind mathematische Verfahren nur bei hoher Datenqualität sinnvoll einsetzbar. Der Einsatz einer linearen ICA zur Signaltrennung brachte unter diesem Gesichtspunkt als Datenvorverarbeitung keinen Vorteil für die Klassifikation der Daten. Weitere Variablen (z.B. Mediatoren), die für diese Arbeit nicht zur Verfügung standen, könnten in Zukunft mit in die Analysen einbezogen werden. Man kann sagen, dass unter den gegebenen Umständen die Möglichkeiten zur Klassifikation und Regelgenerierung auf den vorhandenen Daten ausgeschöpft wurden und nur qualitative Verbesserungen der Datenerhebung eine qualitative Verbesserung der Ergebnisse nach sich ziehen kann.

Für alle drei Datenbasen, die zur Verfügung standen, wurden nach Überprüfung der Leistungsfähigkeit der Verfahren auf Benchmark-Daten Klassifikations- und Regelgergebnisse erzeugt, die Einblicke in die Struktur der Daten des septischen Schocks ermöglichten (Kap. 5 bis 7). Um diese Ergebnisse zu erzielen, waren umfangreiche, individuelle Datenvorverarbeitungen nötig, insbesondere die Fehlerentdeckung und deren Beseitigung, das Sampling der in unregelmäßigen und unterschiedlichen Zeitabständen gemessenen Daten und die Behandlung der fehlenden Werte. Einfache Clusteranalysen führten zu keinem sinnvollen Ergebnis auf den Daten; die Berücksichtigung der Klassen „überlebend“ bzw. „verstorben“ war wesentlich für die erzielten Ergebnisse.

Inwieweit adaptive Datenanalyseverfahren eingesetzt werden können, um Hinweise auf sinnvolle *Therapien* des septischen Schocks zu finden, bleibt z.Z. offen; vermutlich wird für diesen Teilaspekt ein deutlich höherer Anteil medizinischen Expertenwissens integriert werden müssen.

Soweit die medizinische Anwendung; die eingesetzten Verfahren zur Regelgenerierung und Klassifikation sind anwendungsunabhängig und können selbstverständlich auch in anderen Anwendungen eingesetzt werden. Allerdings werden andersartige, individuelle Datenvorverarbeitungen für andere Datenbasen verwendet werden müssen. Eine Regelgenerierung ist und bleibt in Anwendungen besonders wertvoll, da so Erklärungen für den Anwender bereitgestellt werden können, die ihm grundlegende Einblicke ermöglichen und sein Verständnis für die Ergebnisse erhöhen.

Um die Praxistauglichkeit zu gewährleisten, wurde im Kapitel 3 ein Ansatz aus dem Gebiet der adaptiven *Neuro-Fuzzy-Regelgewinnung* weiterentwickelt und für die Anwendung auf metrischen Realweltdaten evaluiert. Das Überlappungsverhalten und der shrink-Mechanismus des Verfahrens wurden verbessert. Interessant war die Untersuchung der Erzeugung scharfer, rechteckiger Regeln zu unterschiedlichen Fuzzy-Niveaus, die gezeigt hat, dass nicht unbedingt dem höchsten Fuzzy-Niveau die statistisch konfidentesten Regeln zugeordnet werden können. Generiert man separat für jede Regel die optimale scharfe Regel, so erreicht man eine deutlich bessere

Konfidenz im Mittel als wenn man die Regeln zu einem festen Fuzzy-Niveau betrachtet. Auf einen Defuzzifizierungsprozess wurde bewusst verzichtet, da die septischen Schock-Daten keine bedeutende linguistische Interpretation zulassen. Die Variante „Fuzzy-Wv-RecBF-DDA“, die einen Gewichtsvektor anstelle eines Zählers verwendet, kann für Daten mit Ausreißern vorteilhaft eingesetzt werden; für Datensätze ohne Ausreißer ergeben sich aber keine Vorteile.

Im 4. Kapitel wurde ein praxistauglicher Regelgenerator mit darauf aufbauenden Klassifikator für symbolische Daten vorgeschlagen (*Generalisierungsregeln*), der auf der mathematischen schichtweisen Durchschnittsbildung basiert. Die Eigenschaften dieses Verfahrens wurden untersucht. Vorteilhaft sind die Optimalität der Regeln bei vollständiger Regelgenerierung und die Möglichkeiten fehlende Werte zu tolerieren, Negationen von Attributen zu verwenden und insbesondere mehrfache, nicht-eindeutige Klassenzugehörigkeiten zu verarbeiten. Um die kombinatorische Explosion der vollständigen Regelgenerierung bei nicht gut zu generalisierenden großen Datensätzen zu vermeiden, wurden Heuristiken wie z.B. Generalisierungsindices eingeführt, die das Ergebnis der Regelgenerierung nicht zum Nachteil veränderten, aber für eine deutlich schnellere Laufzeit der Analysen sorgten. Der Vergleich mit anderen aktuellen Ansätzen wie Assoziationsregeln und FP-Bäumen hat gezeigt, dass der Generalisierungsregelmechanismus nicht zu viele Regeln in einer vergleichbaren Laufzeit mit Speicherplatzvorteilen generiert.

Sowohl für den metrischen als auch den symbolischen Regelgenerator wurde eine Kenngröße angegeben, mit der man die globale *Wichtigkeit* der verwendeten Variablen bzw. Items für die Regelgenerierung bewerten kann. Diese Kenngröße wurde auf Benchmark-Daten evaluiert und konnte wichtige Variablen für das Alarmsystem identifizieren. Mit Hilfe der Wichtigkeit von Variablen kann die Anzahl von Variablen für eine Analyse reduziert werden, so dass für den Anwender einfachere, verständlichere Regeln generiert werden können.

Die eingeführte *Frühzeitigkeit* konnte insbesondere sinnvoll bei den symbolischen septischen Schock-Regeln verwendet werden; so konnten Gruppen von Regeln identifiziert werden, die frühzeitiger oder später auftreten. Bei den metrischen Daten konnten keine speziellen frühzeitigen Regeln identifiziert werden, da die Patientenzustände, die den Regeln zugrunde liegen, zu jeder Zeit der Liegedauer auftreten können.

Grundsätzliche Vergleiche der Verfahren mit anderen wurden vorgenommen (z.B. Fuzzy-RecBF-DDA mit NEFCLASS oder Generalisierungsregeln mit Assoziationsregeln). Dabei stellte sich heraus, dass das (modifizierte) Fuzzy-RecBF-DDA und die Generalisierungsregeln für die Anwendung auf den septischen Schock-Daten geeigneter sind als die jeweiligen Alternativen. Umfangreichere empirische Leistungsvergleiche könnten z.B. Inhalte von Diplomarbeiten werden.

Um statistisch repräsentative Ergebnisse zu erhalten, wurden für die einzelnen Datensätze mehrere Regelmengen auf der Grundlage zufälliger Datenpartitionen (Trainings- und Testdaten) erzeugt. Um aus der Menge der erzeugten Regelmengen eine repräsentative Regelmenge auszuwählen, wurde auf der Basis eines *Ähnlichkeitsmaßes* eine Kenngröße entwickelt, die es erlaubt, eine solche Auswahl zu treffen. Dieses Paradigma wurde sowohl für das Neuro-Fuzzy-System als auch für die Generalisierungsregeln entwickelt und für die Auswahl einer konkreten Regelmenge für das Alarmsystem verwendet. Das Ähnlichkeitsmaß kann auch sinnvoll eingesetzt werden, um Anwendern eine Auswahl von unähnlichen Regeln zu präsentieren, ohne redundante, sehr ähnliche Regeln anzugeben. Das Vorgehen, das individuell auf die jeweiligen Verfahren zugeschnitten ist, lässt sich auf andere Regelgeneratoren übertragen.

Für die Zukunft ist eine weitere Systematisierung und ein Ausbau der Analysemöglichkeiten interessant, z.B. eine Integration der Zeitreihenstruktur oder Zustandsanalysen in die Regelgenerierung oder eine weitere theoretische Untersuchung der Optimalität der Regeln (Widerspruch zwischen geringer Anzahl von Regeln und hoher Klassifikationsleistung bzw. Regelkonfidenz). Prinzipiell können nach entsprechender Vorverarbeitung symbolische Daten mit dem Neuro-Fuzzy-System verarbeitet werden oder umgekehrt metrische Daten mit dem Generalisierungsregel-Verfahren; eine bessere Integration von metrischen und symbolischen Daten wäre wünschenswert, wengleich einer solchen Integration Grenzen gesetzt sind. Speziell für die Klinikdaten wurde in dieser Arbeit eine individuelle Kombination der symbolischen und metrischen Regelmengen vorgenommen, die gezeigt hat, dass allein die metrischen Daten für ein Alarmsystem geeignet sind, da die zusätzlich zu den 3 metrischen Messwerten betrachtete hohe Anzahl von bis zu 35 symbolischen Items nur zu einer sehr geringen Verbesserung des Klassifikationsergebnisses geführt hat. Unter diesen Umständen kann Ärzten nicht zugemutet werden, bis zu 35 zusätzliche symbolische Items anzugeben.

Die Abhängigkeit aller adaptiven Verfahren von Parametereinstellungen (z.B. Häufigkeitsschwellen, Lernrate usw.) bleibt ein offenes Problem. Nimmt man einen strengen Standpunkt ein, so wird ein Verfahren erst dann vollständig *adaptiv*, wenn die Parametereinstellungen anwendungsunabhängig sind. Hier ist bei vielen adaptiven Verfahren noch keine endgültige Lösung gefunden worden.

# Anhang A

## Daten und Analysesoftware

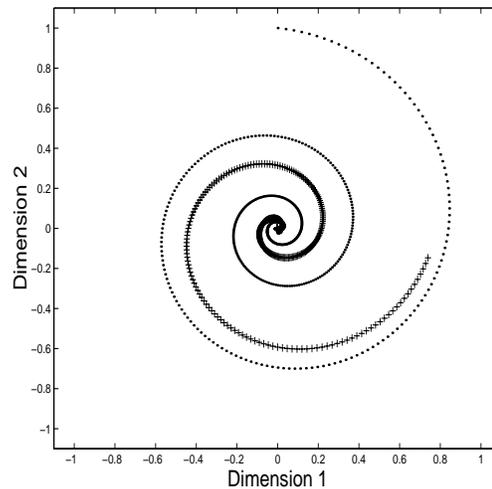
### A.1 Benchmark-Daten

Folgende bekannte Benchmark-Datensätze wurden verwendet:

- Proben-Daten (Diabetes1, Cancer1): verwendet in [Pre94], Diabetes1: 8 metrische Variablen, 2 Klassen, 768 Messungen, Cancer1: 9 metrische Variablen, 2 Klassen, 699 Messungen (<ftp://ftp.informatik.uni-freiburg.de/documents/papers/neuro/proben/proben1/>, Stand: Dez. 2001),
- IRIS-Daten: verwendet in [BH95], 4 metrische Variablen, 3 Klassen, 150 Messungen (<ftp://ftp.ics.uci.edu/pub/machine-learning-databases/iris/>, Stand: Dez. 2001),
- zweidimensionale metrische Spiraldaten: verwendet in [BD98], 2 Klassen, 972 Messungen, siehe Abb. A.1 (bei der inneren Spirale wurden absichtlich einige Datenpunkte weggelassen),
- Vote-Daten: 16 binäre Variablen einer Wählerbefragung mit fehlenden Werten, 2 Klassen, 435 Messungen (<http://yoda.cis.temple.edu:8080/UGAIWWW/lectures/C45/#5>, Stand: Dez. 2001),
- Breast-Daten: 10 binäre Variablen einer Zelluntersuchung, 2 Klassen, 692 Messungen (<http://www.phil.gu.se/ANN/datasets.html#breast>, Stand: Dez. 2001),
- Benchmark-Datensatz mit Ausreißern: 2 Klassen, 400 Messungen (davon 10 Ausreißer), siehe Abb. A.2 (nicht aus einem Datenarchiv, selbst erzeugt).

### A.2 Vorstudien Daten

In diesem Abschnitt werden die Daten der Vorstudie beschrieben und anschließend kurz erläutert. Allgemein von *MEDAN-Daten* wird gesprochen, wenn nicht zwischen



**Abbildung A.1:** Spiraldaten, + Klasse 1, · Klasse 2.

IDVar	Variable	Einheit	NMin	NMax	GMin	GMax
1	Herzfrequenz	/min	50	100	0	500
2	Syst. Blutdruck	mmHg	80	180	0	500
3	Temperatur	°C	36.5	38	20	50
4	Leukozyten	1000/ $\mu$ l	4.4	11.3	0	100

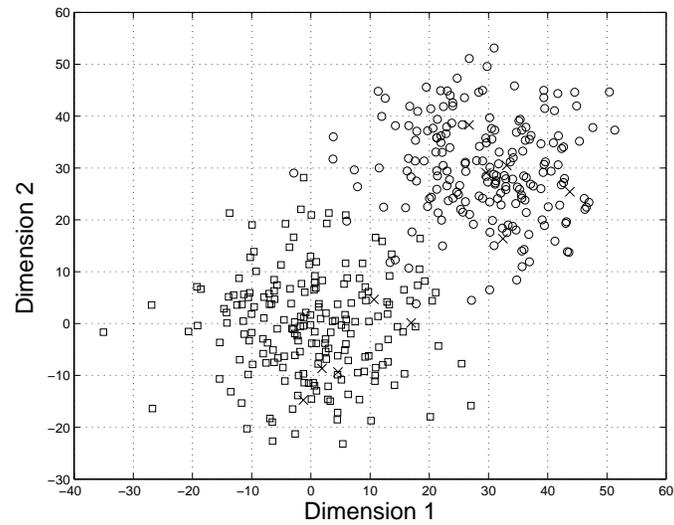
**Tabelle A.1:** Trenddaten der Vorstudie.

den Daten der Vorstudie, die sich aus zwei Studien zusammengesetzt hat, den ASK-Daten und den Klinikdaten unterschieden wird.

Die relevanten Aufnahme- und Entlassungsdaten (Patienten-ID, Alter, Größe, Gewicht, Aufnahme- und Entlassungsdatum, Verstorben) enthält die Tabelle PATIENT.

Die Tabelle TRENDDATEN enthält die Schlüssel, die Einheiten, die Normwerte (Minimum und Maximum NMin bzw. NMax) und die Grenzwerte (Minimum und Maximum GMin bzw. GMax) für die einzelnen Messwerte. Einige Beispiele der 133 Variablen sind in der Tab. A.1 eingetragen. Dabei ist zu berücksichtigen, dass von den 133 Variablen in den beiden Vorstudien „nur“ 63 mit einer grundsätzlich nicht mangelhaften Qualität dokumentiert wurden. Diese Variablen besitzen aber teilweise noch eine große Anzahl von fehlenden Werten (siehe das Kapitel 5).

Anschließend sei noch ein Ausschnitt aus der Tabelle TRENDDATEN\_DER\_PATIENTEN in der Tab. A.2 wiedergegeben, in der die Messwerte mit Zeitstempel stehen (insgesamt etwa 120000 Eintragungen). Dabei ist zu beachten, dass in der



**Abbildung A.2:** Normalverteilte Daten mit Ausreißern. Viereck: Klasse 1, Kreis: Klasse 2, Kreuz: Ausreißer der jeweils anderen Klasse

IDPat	IDVar	Wert	Datum	Uhrzeit
3	1	120	8.11.93	8:00:00
3	1	100	8.11.93	12:00:00
3	2	110	8.11.93	8:00:00
4	1	80	12.11.93	8:00:00

**Tabelle A.2:** Zeitstempel der Trenddaten der Vorstudie.

Vorstudie die Uhrzeiten nur zur Unterscheidung der verschiedenen Samples eines Patienten an einem Tag dienen; sie entsprechen nicht wie bei den Klinikdaten den echt gemessenen Uhrzeiten.

**A.2.1 Bemerkung: (Subsysteme)**

Neben den *Stammdaten* (Alter, Geschlecht, Größe, Gewicht) existieren folgende Subsysteme. Zunächst die Tabelle für die Subsysteme der gemessenen Werte; Mehrfachzuordnungen sind möglich:

- Herz/Kreislauf: Herzfrequenz, systolischer, diastolischer und mittlerer Blutdruck, ZVD, KreatininKinase, Cardiac Index, PAPm, PAWP, SVRI, PVRI, SvO<sub>2</sub>, O<sub>2</sub>AVI, VO<sub>2</sub>l, O<sub>2</sub>Eptr, Qs/Qt
- Swan-Ganz-Katheter: Cardiac Index, PAPm, PAWP, SVRI, PVRI, SvO<sub>2</sub>, O<sub>2</sub>AVI, VO<sub>2</sub>l, O<sub>2</sub>Eptr, Qs/Qt
- Lunge/Beatmung: pO<sub>2</sub>, pCO<sub>2</sub>, BaseExcess, NaHCO<sub>2</sub>, O<sub>2</sub>-Sättigung, Peak, I:E, Beatmungsform, O<sub>2</sub>-Konzentration Inspiratorisch, Beatmungsfrequenz
- Säure-Base-Haushalt: pO<sub>2</sub>, pCO<sub>2</sub>, BaseExcess, NaHCO<sub>2</sub>, pH-Wert, Kalium
- Infektion: Temperatur, CRP, Procalcitonin, Leukozyten, Eisen, Thrombozyten, Fibrinogen, D-Dimere, Reptilase, AT III
- Blutbild: Leukozyten, Eisen, Erythrozyten, Hämoglobin, Hämatokrit
- Gerinnung: Thrombozyten, Fibrinogen, D-Dimere, Reptilase, TPZ, PTT, TZ, AT III
- Leber: TPZ, PTT, TZ, AT III, GOT, GPT, GGT, Bilirubin gesamt, Bilirubin direkt, Bilirubin indirekt, CHE, GLDH, NH<sub>3</sub>, LDH, Cholesterin, Triglyzeride, Alkalische Phosphata, Laktat, Amylase, Lipase, Albumin, Gesamtprotein, KOD
- Pankreas: Amylase, Lipase
- Blutung: Albumin, Gesamtprotein, KOD
- Niere: Harnsäure, Harnstoff, Kreatinin, U-Kalium, U-Natrium, U-Chlor, U-Kreatinin, U-spezifisches Gewicht, Urinmenge, Kalium
- Vielfachanalyse/Elektrolyte: Kalium, Natrium, Calcium, Chlorid, Phosphat
- sonstige gemessene Werte: Blutzucker, Magensondenmenge/Erbrechen

Anschließend werden die Subsysteme für die Medikamente und die Therapien aufgelistet; dabei sind einige Bezeichnungen Oberbegriffe, die für eine Vielzahl verschiedener Medikamente stehen können:

- Opioide: Opioide
- Katecholamine: Noradrenalin, Suprarenin (=Adrenalin), Dopamin hochdosiert, Dopamin niedrigdosiert, Dobutrex (=Dobutamin), Dopexamin

- Blutdrucksenker: Blutdrucksenker
- Antiarrythmika: Antiarrythmika
- Narkotika/Analgetika: Ketamin, Midazolam
- Volumengabe: Kristalloid, HAES
- Diuretika: Furosemid
- Brocholytika/Antiasthmatica: Theophyllin
- Blutprodukte: Thrombozytenkonzentration, FFP/PPSB, EK-Gabe
- Gerinnung: Thrombozytenkonzentration, FFP/PPSB
- Immunsystem: Chemotherapie, Kortikoide
- Lunge/Beatmung: O<sub>2</sub>-Gabe, PEEP
- Niere: Hämofiltration, chronische Dialyse
- Leber: Hämofiltration
- sonstige Medikamente/Therapien: physische Kühlung, Antipyretika, Antibiotika, Insulin, Alprostadil, Flolan, Antikoagulation

Zum Schluss werden die Subsysteme für die Diagnosen wiedergegeben:

- Lunge/Beatmung: ARDS
- Operation: Trauma, Leberresektion
- Zustand: Infektion, Sepsis, Septischer Schock
- Leber: Leberzirrhose, Lebererkrankung, akutes Leberversagen, Cholestase
- Herz/Kreislauf: akute Herzerkrankung, Herzinfarkt, akutes Rechtsherzversagen, Massive Blutung
- Blutbild: Splenektomie, Hämolyse
- Infektion: DIC
- Pankreas: Akute Pankreatitis, chronische Pankreatitis
- Niere: Niereninsuffizienz
- Gastrointestinal-Trakt: Mesenterialinfarkt
- weitere Diagnosen: HIT, Diabetes

Die verwendeten Datensätze, die sich aus den hier aufgeführten Daten zusammensetzen, wurden bei jeder Analyse näher beschrieben. – Zu beachten ist, dass nicht alle der oben aufgeführten Variablen in der Vorstudie verfügbar waren, insbesondere die Diagnosen, viele Medikamente, die Daten des Swan-Ganz-Katheters und die Beatmungsform waren *nicht* vorhanden bzw. auswertbar (z.B. wegen handschriftlichen unvollständigen Eintragungen in der Datenbank mit unterschiedlichen Abkürzungen).

### A.3 Klinikdaten

Die Datenbank für die Kliniken wurde aufgrund der Erfahrungen mit den Vorstudien-daten vollständig überarbeitet. Im August 2001 enthielt die Datenbank 138 dokumentierte Patientenakten. Einige Variablen sind weggefallen, einige sind hinzugekommen. In diesem Anhang wird eine Übersicht der Variablen gegeben. Die Datentypen sind für die Trenddaten Real-Werte, für die meisten Medikamente, Diagnosen und Therapien binäre Werte oder Integer-Klassen (z.B. 1 = Lebertransplantation, 2 = Herztransplantation, usw.). Zu beachten ist, dass auch hier nicht alle Variablen vollständig zur Verfügung standen; die zur Analyse verwendeten Datensätze sind bei der jeweiligen Analyse beschrieben.

**Stammdaten:** Alter, Geschlecht, Größe, Gewicht, Aufnahme-datum, Entlassungsdatum, ExitusLetalis, Therapia Minima, Hauptdiagnose (Einlieferungsgrund), Liegedauer (errechnet), Body Mass Index (errechnet).

**Trenddaten:** Herzfrequenz, systolischer Blutdruck, diastolischer Blutdruck, mittlerer Blutdruck (errechnet), Temperatur, ZVD, PCWP=PAWP=PAOP, pH-Wert, pO<sub>2</sub> arteriell, pCO<sub>2</sub> arteriell, Base Excess, Natriumbicarbonat, O<sub>2</sub>-Sättigung, Leukozyten, Erythrozyten, Hämoglobin, Hämatokrit, Thrombozyten, TPZ, PTT, TZ, AT3, Fibrinogen, Natrium / Serum, Kalium / Serum, Calcium / Serum, Chlorid / Serum, Kreatinin / Serum, Harnstoff, Harnsäure, GOT (ASAT), GPT (ALAT), GGT, Alkalische Phosphatase, Bilirubin gesamt, LDH, Cholesterin, Triglyceride, Albumin, Gesamtprotein, CRP, Eisen, Blutzucker, Lungeninfiltration, Urinmenge.

**Medikamente:** Analgetika, Antiarrhythmika, Antibiotika, Antihypertonika, Antihypotonika, Antikoagulation, Antimykotika, Antipyretika, THAM / Tris, Natriumbicarbonat (Einfuhr), Broncholyse, Diuretika, Schleifendiuretika, Immunglobuline / Polyglobulin, Pentaglobin, Immunsuppression (Kortikoide), Kristalloide, Kolloide, EK, FFP, Leukozytenkonzentrat, Thrombozyten, Albumin (5% und 20%, Einfuhr), PPSB, AT3 (Einfuhr), einzelne Gerinnungsfaktoren, G-CSF, Insulin, Adrenalin (Perfusor oder Eimalgabe), Noradrenalin (Perfusor oder Eimalgabe), Dopamin (Perfusor oder Eimalgabe), Dobutamin (Perfusor oder Eimalgabe), Muskelrelaxantien, Sedativa / Narkotika.

**Therapien:** O<sub>2</sub>-Gabe, Peak, I:E, Beatmungsform, Beatmungsfrequenz, FiO<sub>2</sub>, PEEP, Hämofiltration, Dialyse, Organtransplantation, Katheter.

**Ernährung:** Parenterale Ernährung, enterale Ernährung.

**Diagnosen:** Diabetes mellitus, akute Herzerkrankung (Herzinfarkt, Rhythmusstörung, akute Herzinsuffizienz, sonstige), chronische Herzerkrankung (Herzinsuffizienz, KHK, Hypertonus, Rhythmusstörungen, sonstige), akutes Nierenversagen, chronische Nierenerkrankung (Dialyse, Niereninsuffizienz, sonstige), akute Lungenerkrankung (ARDS, Pneumonie, sonstige), chronische Lungenerkrankung (Asthma, COPD, sonstige), chronische Lebererkrankung, Pankreatitis (akut oder chronisch), Immunsuppression (Leukämie, Lymphom, Bestrahlung, Chemotherapie, sonstige), HIT.

**Operationen:** alle Operationen gemäß der OP301-Codierung.

**Scores:** SOFA, APACHE II, SAPS II, MODS, SepsisScoreVariable (alle Scores wurden bei fehlenden Werten modifiziert errechnet bzw. als nicht berechenbar deklariert).

## A.4 Ausgewählte symbolische Klinikdaten

Da zum einen die Klinikdaten aus dem Abschnitt A.3 erst im Sommer 2001 zur Verfügung standen und andererseits eine der Daten liefernden Kliniken bereits symbolische Variablen einer größeren Anzahl von Patienten anbot, wurden die symbolischen Daten aus den Tabellen A.3 und A.4 separat analysiert. Alle Variablen wurden als binäre Variablen verwendet. Die geraden Zahlen  $n$  wurden verwendet, um die Negationen der Variablen mit den ungeraden Zahlen  $n - 1$  zu codieren. Abgekürzt seien die ausgewählten symbolischen Klinikdaten als *ASK-Daten* bezeichnet.

## A.5 Analysesoftware

Fast alle verwendeten Programme wurden mit der Programmierumgebung Matlab (Version 5.3) erstellt. Die Trainingsalgorithmen der Regelgenerierung wurden in C (Microsoft Visual C 6.0) implementiert, um eine um Größenordnungen deutlich bessere Laufzeit zu erhalten. Die auf der S. 272 genannten „Programmgruppen“ wurden von mir implementiert. Der FastICA-Algorithmus von Hyvärinen wurde aus dem Internet bezogen unter (<http://www.cis.hut.fi/projects/ica/fastica/index.html>, Stand: Dezember 2001). Die von Borgelt erstellte Implementierung eines Assoziationsregelmechanismus, die im Abschnitt 6.6 verwendet wurde, ist erhältlich unter (<http://fuzzy.cs.uni-magdeburg.de/~borgelt/software.html>, Version 2.11, Stand:

Nr.	Variable
1	Organinsuffizienz
3	Traumatisch
5	GCS = 15
7	Metastatischer Tumor
9	Bösartiges Haematom
11	Diabetes mellitus präoperativ
13	Diabetes mellitus insulinpflichtig
15	Leberzirrhose präoperativ
17	Erythrozytengabe
19	Plasmagabe
21	Thrombozytengabe
23	SOFA (Score) > 12
25	Multiorgan-Score > 3
27	Anzahl der Organausfälle > 2
29	Parentale Ernährung
31	Enterale Ernährung
33	Steroide
35	Beatmung
37	PAK
39	Hämofiltration
41	Dialyse
43	Peritonealspülung
45	Reanimation
47	Cardioversion

**Tabelle A.3:** Variablen der ausgewählten symbolischen Klinikdaten I. Variablen 1 bis 48. Die Negationen der Variablen mit der Nr.  $n$  werden durch die ungeraden Nummern  $n + 1$  kodiert.

Nr.	Variable
49	Schrittmacher
51	Antiarrythmika-Gabe
53	Volumengabe um RR > 100
55	Hypotension trotz Substitution
57	Bronchoskopie
59	Relaxierung
61	Endoskopie
63	Volle enterale Ernährung
65	Auffälligkeit beim Gastrointestinaltrakt
67	Cholecystitis (keine Steine)
69	Stressulkus
71	nekrotisierende Enterocolitis
73	nekrotisierende Pankreatitis
75	spontane Gallenblasenperforation
77	akutes Nierenversagen
79	Hämodialyse
81	hämorrhoidale Diathese
83	mind. 3 Antibiotika gegeben
85	Infektion
87	Eiter (abgeschlossener Raum)
89	Infektionsquelle nachgewiesen
91	Druckinfusion
93	Reoperation
95	Katecholamingabe

**Tabelle A.4:** Variablen der ausgewählten symbolischen Klinikdaten II. Variablen 49 bis 96. Die Negationen der Variablen mit der Nr.  $n$  werden durch die ungeraden Nummern  $n + 1$  kodiert.

Dez. 2001). Mit dem Programm „SOM Toolbox Version 1.0 beta“ (<http://www.cis.hut.fi/projects/somtoolbox/download/>, Stand: Dez. 2001) wurden U-Matrizen erstellt.

### **Toolbox Metrische Regelgenerierung, Kombination und Alarmsystem:**

In dieser Toolbox ist das *Fuzzy-RecBF-DDA-Netz* inkl. der Verbesserungen zur metrischen Regelgenerierung implementiert sowie die Algorithmen zur Gütebewertung, zur Wichtigkeit und zur Modellauswahl aus dem Kapitel 3. Auf der Basis der Regelmengen ist es möglich, Alarme zu generieren und deren Qualität zu analysieren; diese Module sind ebenfalls hier zusammengefasst. Zusätzlich sind Module zu den Versuchen der Kombination symbolischer und metrischer Regeln vorhanden. Zu dieser Toolbox gehören auch die notwendigen Module zur Durchführung der Experimente auf den Anwendungsdaten.

### **Toolbox Symbolische Regelgenerierung:**

Hier ist der *Generalisierungsregelmechanismus* aus dem Kap. 4 inklusive den vorgestellten heuristischen Erweiterungen implementiert. Ebenso sind die Gütebewertungen und die für die Anwendungen notwendigen Module in der Toolbox zusammengestellt.

### **Toolbox Datenvorverarbeitung:**

Neben einfachen mathematischen (z.B. Normierungen) oder wahrscheinlichkeitstheoretischen Modulen (z.B. ROC-Kurve) war die Implementierung von Modulen zur Datenvorverarbeitung nötig:

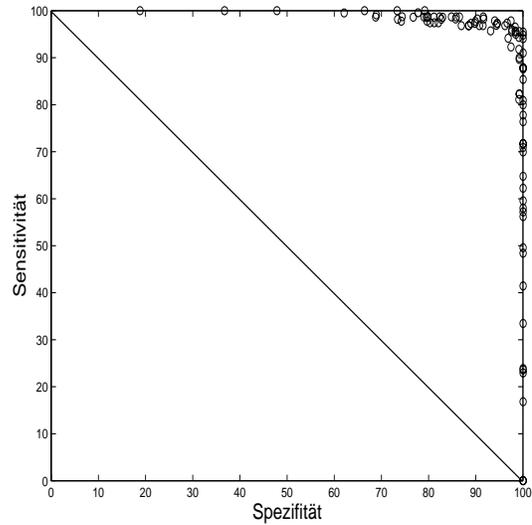
1. FormatTimeSeries, FormatClassification, FormatRule (Formatierungen),
2. SelectPatient, SelectVariable, SelectTime (Auswahl und Gruppierung),
3. BorderCheck, MissingValueRemoval (Grenzwert- und fehlende Werte-Analyse),
4. DynamicCheck (Zeitreihendynamikanalyse),
5. Sampling mit wahlweiser Samplingrate, -methode.

# Anhang B

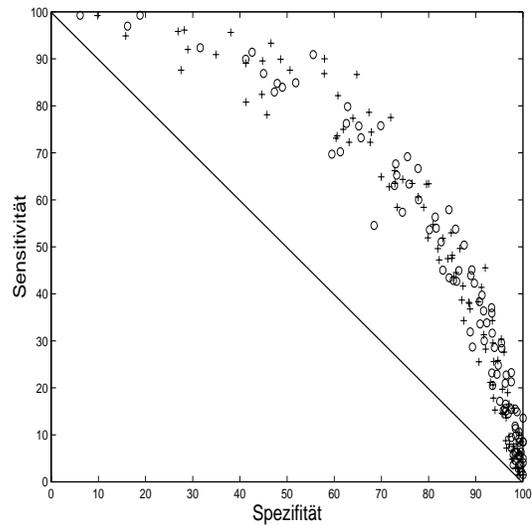
## ROC-Kurven

In diesem Anhang sind die ROC-Kurven für die folgenden Datensätze wiedergegeben:

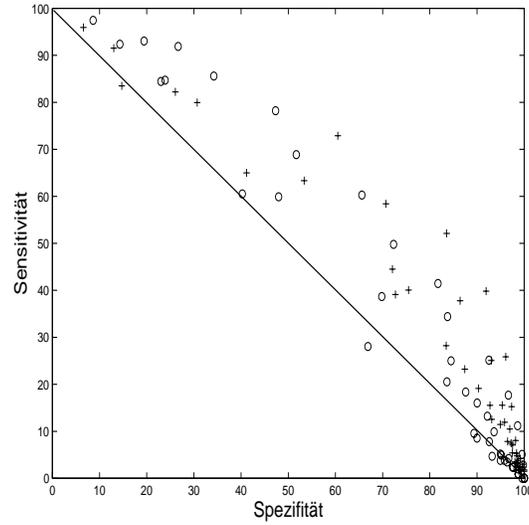
- Cancer1,
- Diabetes1 (mit und ohne LICA),
- Häufig12Vor zusammen mit Wichtig17Vor,
- Häufig16K,
- SOFA1K,
- SystThrombo2K,
- Wichtig6K,
- SystDiasThrombo3K und
- SystDiasThrombo3K/Sym35.



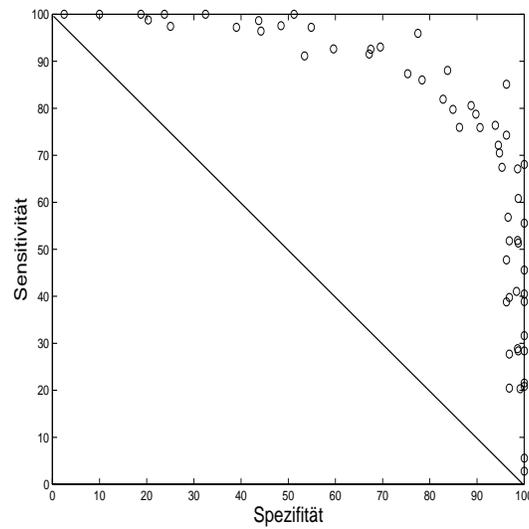
**Abbildung B.1:** Cancer1-Daten: ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche aller 10 Versuchswiederholungen. Die Diagonale ist als Linie eingezeichnet. Angaben in %.



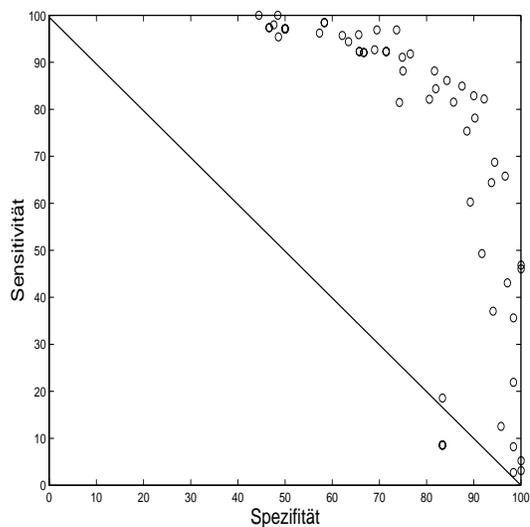
**Abbildung B.2:** Diabetes1-Daten: ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche jeweils aller 10 Versuchswiederholungen ohne ICA (o) und mit ICA (+). Die Diagonale ist als Linie eingezeichnet. Angaben in %.



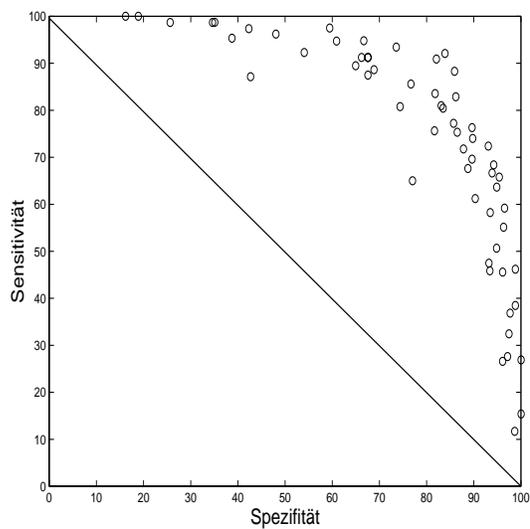
**Abbildung B.3:** Häufig12Vor-Daten (+) und Wichtig17Vor-Daten (o): ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche der jeweils 5 Versuchswiederholungen. Die Diagonale ist als Linie eingezeichnet.



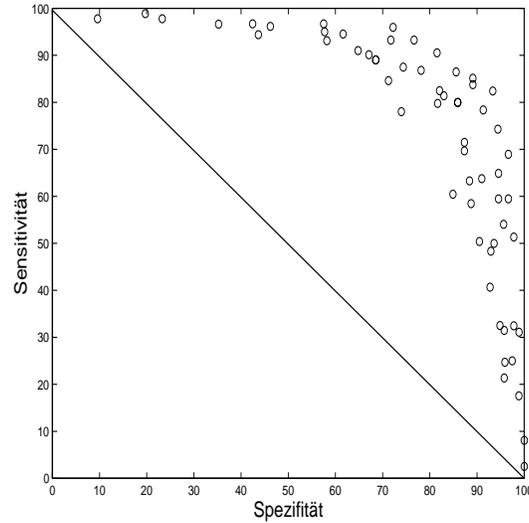
**Abbildung B.4:** Häufig16K-Daten: ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche aller 5 Versuchswiederholungen. Die Diagonale ist als Linie eingezeichnet. Angaben in %.



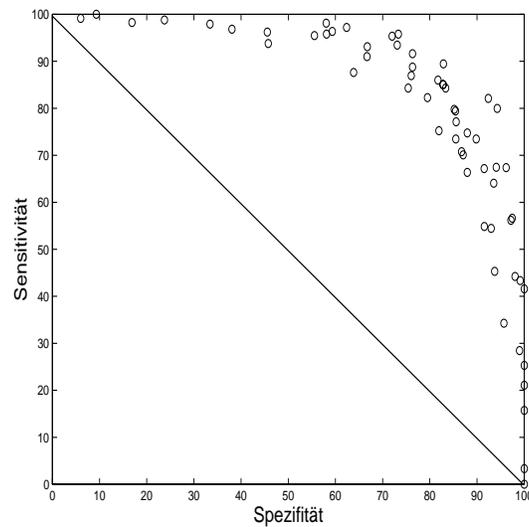
**Abbildung B.5:** SOFA1K-Daten: ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche aller 5 Versuchswiederholungen. Die Diagonale ist als Linie eingezeichnet. Angaben in %.



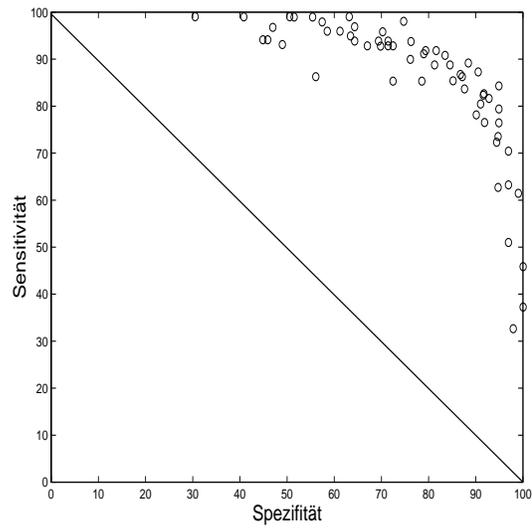
**Abbildung B.6:** SystThrombo2K-Daten: ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche aller 5 Versuchswiederholungen. Die Diagonale ist als Linie eingezeichnet. Angaben in %.



**Abbildung B.7:** Wichtig6K-Daten: ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche aller 5 Versuchswiederholungen. Die Diagonale ist als Linie eingezeichnet. Angaben in %.



**Abbildung B.8:** SystDiasThrombo3K-Daten: ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche aller 5 Versuchswiederholungen. Die Diagonale ist als Linie eingezeichnet. Angaben in %.



**Abbildung B.9:** SystDiasThrombo3K/Sym35-Daten: ROC-Kurven (ohne interpolierende Verbindungslinien) der Klassifikationsversuche aller 5 Versuchswiederholungen. Die Diagonale ist als Linie eingezeichnet. Angaben in %.

# Anhang C

## Tabellen

In diesem Anhang sind als Erstes die Konfidenz- und Häufigkeitswerte der Core- und Support-Regeln aus den Abschnitten 3.2.1 bis 3.4, 3.6, 5.5 und 7.2.2 für die folgenden Datensätze wiedergegeben:

- IRIS,
- Cancer1,
- Diabetes1,
- Spiraldaten (ohne *nearby*-Parameter),
- Spiraldaten (mit *nearby*-Parameter),
- Diabetes1 (mit fehlenden Werten),
- NormAusreißer-Daten,
- Häufig12Vor,
- Wichtig17Vor,
- Häufig16K,
- SOFA1K,
- SystThrombo2K,
- Wichtig6K und
- SystDiasThrombo3K.

Ab Tab. C.31 folgen die Tabellen der Alarmkennzahlen aus dem Abschnitt 7.5.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	100.00	0.00	100.00	100.00
Klasse 2 (Core)	94.38	10.31	66.67	100.00
Klasse 3 (Core)	86.92	14.48	58.99	100.00
Klasse 1 (Support)	99.03	3.06	90.32	100.00
Klasse 2 (Support)	74.20	14.23	45.93	96.97
Klasse 3 (Support)	77.92	17.23	43.23	95.83

**Tabelle C.1:** IRIS-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	27.07	4.08	18.67	30.67
Klasse 2 (Core)	18.18	8.22	2.22	30.67
Klasse 3 (Core)	16.18	6.05	6.67	26.67
Klasse 1 (Support)	33.87	5.00	24.00	41.33
Klasse 2 (Support)	36.00	10.02	22.22	48.44
Klasse 3 (Support)	28.97	8.79	20.00	50.67

**Tabelle C.2:** IRIS-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	85.86	17.86	41.67	99.19
Klasse 2 (Core)	50.60	21.02	24.56	92.75
Klasse 1 (Support)	67.88	10.84	56.66	88.48
Klasse 2 (Support)	46.92	17.59	26.62	78.76

**Tabelle C.3:** Cancer1-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	17.49	8.99	2.19	32.21
Klasse 2 (Core)	27.12	10.64	16.12	46.43
Klasse 1 (Support)	40.58	10.44	25.09	62.00
Klasse 2 (Support)	33.50	10.77	19.71	52.57

**Tabelle C.4:** Cancer1-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	52.50	9.77	34.28	66.29
Klasse 2 (Core)	31.76	9.07	17.78	45.83
Klasse 1 (Support)	63.60	3.55	58.56	70.89
Klasse 2 (Support)	40.84	7.14	32.14	53.92

**Tabelle C.5:** Diabetes1-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	4.37	1.46	1.72	6.88
Klasse 2 (Core)	1.65	1.29	0.29	5.03
Klasse 1 (Support)	17.64	3.96	13.39	27.39
Klasse 2 (Support)	13.57	5.57	6.14	25.37

**Tabelle C.6:** Diabetes1-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	36.54	11.92	12.44	50.32
Klasse 2 (Core)	47.12	17.80	25.56	82.20
Klasse 1 (Support)	50.92	4.15	45.93	59.86
Klasse 2 (Support)	52.88	5.11	45.62	58.78

**Tabelle C.7:** Spiraldaten ( $nearby = 2$  (ohne)): Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	5.74	5.53	1.30	18.86
Klasse 2 (Core)	11.80	8.26	4.26	24.53
Klasse 1 (Support)	36.60	5.35	31.62	44.96
Klasse 2 (Support)	38.96	10.47	25.95	53.25

**Tabelle C.8:** Spiraldaten ( $nearby = 2$  (ohne)): Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	73.96	6.72	61.64	83.31
Klasse 2 (Core)	80.96	9.09	64.82	90.54
Klasse 1 (Support)	48.47	1.46	46.43	51.01
Klasse 2 (Support)	54.14	1.95	51.58	57.11

**Tabelle C.9:** Spiraldaten ( $nearby = 0.14$ ): Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	1.86	0.46	1.40	2.83
Klasse 2 (Core)	3.79	1.42	1.95	6.22
Klasse 1 (Support)	36.13	4.88	31.30	47.10
Klasse 2 (Support)	41.65	10.39	30.41	58.39

**Tabelle C.10:** Spiraldaten ( $nearby = 0.14$ ): Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	47.37	9.70	23.43	56.52
Klasse 2 (Core)	27.95	8.15	20.07	45.58
Klasse 1 (Support)	65.37	3.67	60.02	71.36
Klasse 2 (Support)	38.61	3.27	34.40	44.17

**Tabelle C.11:** Diabetes1-Daten (fehlende Werte): Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	2.71	0.94	1.61	4.55
Klasse 2 (Core)	1.26	0.49	0.58	2.18
Klasse 1 (Support)	15.57	3.48	11.21	23.48
Klasse 2 (Support)	15.79	4.17	9.39	23.19

**Tabelle C.12:** Diabetes1-Daten (fehlende Werte): Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	48.26	9.36	36.76	64.88
Klasse 2 (Core)	48.63	2.03	23.10	72.42
Klasse 1 (Support)	38.26	9.42	26.55	55.65
Klasse 2 (Support)	47.53	12.23	27.35	60.07

**Tabelle C.13:** NormAusreißer-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent, Fuzzy-RecBF-DDA.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	14.03	6.19	4.10	25.00
Klasse 2 (Core)	11.82	5.68	2.50	20.50
Klasse 1 (Support)	37.82	9.03	2.63	52.50
Klasse 2 (Support)	34.89	8.96	2.31	52.00

**Tabelle C.14:** NormAusreißer-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent, Fuzzy-RecBF-DDA.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	60.34	15.59	30.37	77.55
Klasse 2 (Core)	61.53	17.42	18.79	79.29
Klasse 1 (Support)	47.13	9.95	25.52	61.24
Klasse 2 (Support)	46.02	13.80	16.61	61.92

**Tabelle C.15:** NormAusreißer-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent, Fuzzy-WvRecBF-DDA.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse 1 (Core)	15.62	6.53	3.70	24.70
Klasse 2 (Core)	17.17	4.95	9.90	24.63
Klasse 1 (Support)	42.65	9.91	28.50	61.50
Klasse 2 (Support)	40.50	13.45	25.90	63.70

**Tabelle C.16:** NormAusreißer-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent, Fuzzy-WvRecBF-DDA.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	59.40	6.03	52.26	67.61
Klasse v (Core)	8.82	3.33	4.13	13.05
Klasse ü (Support)	65.83	5.58	60.39	75.09
Klasse v (Support)	31.97	0.40	31.30	32.27

**Tabelle C.17:** Häufig12Vor-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	2.18	1.05	1.22	3.69
Klasse v (Core)	0.57	0.57	0.23	1.57
Klasse ü (Support)	11.60	2.81	8.90	15.95
Klasse v (Support)	8.78	2.84	6.17	11.95

**Tabelle C.18:** Häufig12Vor-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	59.96	12.88	41.82	75.63
Klasse v (Core)	11.70	7.56	2.55	19.31
Klasse ü (Support)	74.37	13.48	50.74	84.51
Klasse v (Support)	30.88	15.36	16.50	56.79

**Tabelle C.19:** Wichtig17Vor-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	2.39	1.13	1.36	4.05
Klasse v (Core)	0.62	0.39	0.24	1.23
Klasse ü (Support)	15.93	2.06	13.11	18.63
Klasse v (Support)	13.33	2.79	8.67	15.81

**Tabelle C.20:** Wichtig17Vor-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	56.51	11.43	41.84	70.00
Klasse v (Core)	47.73	21.16	26.57	80.48
Klasse ü (Support)	59.74	8.21	53.13	73.83
Klasse v (Support)	64.48	6.45	57.55	71.81

**Tabelle C.21:** Häufig16K-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	3.00	1.98	1.74	6.51
Klasse v (Core)	2.80	1.17	0.87	3.89
Klasse ü (Support)	31.39	6.78	26.19	43.18
Klasse v (Support)	34.09	11.65	18.16	45.09

**Tabelle C.22:** Häufig16K-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	61.66	19.31	43.48	89.47
Klasse v (Core)	81.07	15.61	56.11	95.46
Klasse ü (Support)	41.24	12.16	26.32	55.12
Klasse v (Support)	59.82	10.03	45.24	69.06

**Tabelle C.23:** SOFA1K-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	21.40	11.61	12.10	41.61
Klasse v (Core)	33.31	14.57	20.00	50.74
Klasse ü (Support)	55.27	24.57	34.07	97.81
Klasse v (Support)	62.84	19.65	32.59	77.37

**Tabelle C.24:** SOFA1K-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	60.10	25.09	23.06	89.62
Klasse v (Core)	41.30	5.92	35.81	51.42
Klasse ü (Support)	63.15	10.44	53.25	79.21
Klasse v (Support)	34.45	9.81	25.23	46.30

**Tabelle C.25:** SystThrombo2K-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	10.57	4.13	5.16	16.41
Klasse v (Core)	8.18	5.85	3.39	17.53
Klasse ü (Support)	37.61	8.94	26.62	51.23
Klasse v (Support)	31.08	5.07	24.89	37.63

**Tabelle C.26:** SystThrombo2K-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	62.37	10.73	49.47	79.21
Klasse v (Core)	48.56	17.32	29.69	72.64
Klasse ü (Support)	63.89	3.79	59.58	69.43
Klasse v (Support)	53.88	6.85	46.22	60.00

**Tabelle C.27:** Wichtig6K-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	7.49	2.82	5.22	10.91
Klasse v (Core)	8.87	6.24	2.50	15.75
Klasse ü (Support)	27.40	6.47	20.38	37.20
Klasse v (Support)	29.11	14.71	13.45	46.25

**Tabelle C.28:** Wichtig6K-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	53.00	16.06	34.97	74.06
Klasse v (Core)	42.08	14.67	20.09	59.19
Klasse ü (Support)	55.80	10.07	44.95	70.54
Klasse v (Support)	39.69	4.94	32.07	45.70

**Tabelle C.29:** SystDiasThrombo3K-Daten: Konfidenz der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	Mittelwert	Standardabw.	Minimum	Maximum
Klasse ü (Core)	7.30	2.61	4.98	11.73
Klasse v (Core)	6.62	1.82	4.53	8.60
Klasse ü (Support)	29.42	14.29	16.11	49.58
Klasse v (Support)	26.66	3.28	23.27	31.28

**Tabelle C.30:** SystDiasThrombo3K-Daten: Häufigkeit der Core- und Support-Regeln pro Klasse auf den Testdaten (Mittelwert, Standardabweichung, Minimum, Maximum). Angaben in Prozent.

	1	2	3	4
verstorben Stufe 1	37.63	29.50	39.04	47.85
überlebt Stufe 1	38.02	32.17	5.34	4.69
verstorben Stufe 2	18.82	16.70	29.66	30.11
überlebt Stufe 2	6.25	3.91	0.00	0.00
verstorben insgesamt	56.45	46.20	68.69	77.96
überlebt insgesamt	44.27	36.07	5.34	4.69

**Tabelle C.31:** Häufig16K-Daten: Alarmkennzahlen. Erklärung siehe Abb. 7.22. Angaben in Prozent.

	1	2	3	4
verstorben Stufe 1	53.54	51.99	37.78	39.39
überlebt Stufe 1	44.91	43.52	16.71	9.72
verstorben Stufe 2	16.16	18.84	36.06	39.39
überlebt Stufe 2	12.96	13.06	0.00	0.00
verstorben insgesamt	69.70	70.82	73.84	78.79
überlebt insgesamt	57.87	56.58	16.71	9.72

**Tabelle C.32:** SOFA1K-Daten: Alarmkennzahlen. Erklärung siehe Abb. 7.23. Angaben in Prozent.

	1	2	3	4
verstorben Stufe 1	59.90	51.26	55.34	65.63
überlebt Stufe 1	33.85	31.07	8.98	7.81
verstorben Stufe 2	5.21	9.91	14.77	14.58
überlebt Stufe 2	7.81	6.02	0.63	0.00
verstorben insgesamt	65.10	61.17	70.11	80.21
überlebt insgesamt	41.67	37.09	9.60	7.81

**Tabelle C.33:** SystThrombo2K-Daten: Alarmkennzahlen. Erklärung siehe Abb. 7.24. Angaben in Prozent.

	1	2	3	4
verstorben Stufe 1	35.06	33.71	51.11	56.90
überlebt Stufe 1	32.41	28.83	5.13	3.24
verstorben Stufe 2	12.07	14.31	21.26	20.12
überlebt Stufe 2	2.78	3.13	0.79	2.78
verstorben insgesamt	47.13	48.02	72.37	77.01
überlebt insgesamt	35.19	31.96	5.92	6.02

**Tabelle C.34:** Wichtig6K-Daten: Alarmkennzahlen. Erklärung siehe Abb. 7.25. Angaben in Prozent.

	1	2	3	4
verstorben Stufe 1	36.87	31.25	34.46	35.86
überlebt Stufe 1	28.70	17.49	5.22	5.56
verstorben Stufe 2	14.65	14.13	27.56	41.41
überlebt Stufe 2	6.48	3.10	0.98	1.85
verstorben insgesamt	51.52	45.37	62.01	77.27
überlebt insgesamt	35.19	20.59	6.20	7.41

**Tabelle C.35:** SystDiasThrombo3K-Daten: Alarmkennzahlen. Erklärung siehe Abb. 7.26. Angaben in Prozent.



# Anhang D

## Das Fuzzy-RecBF-DDA-Netz von Huber/Berthold

In diesem Anhang wird der Originalalgorithmus Fuzzy-RecBF-DDA [HB95] inkl. dem shrink-Algorithmus wiedergegeben. Die Schreibweisen werden aus dem Kap. 3 übernommen, ohne den Algorithmus zu verändern. Anschließend wird noch die Situation aus dem ersten Punkt (S. 69) genauer ausgeführt.

### D.0.1 Algorithmus (Fuzzy-RecBF-DDA, Original):

Die Schritte beschreiben den Lernvorgang einer Epoche.

1. Gewichte zurücksetzen:

```
for l = 1 to c
  for i = 1 to ml
    wil := 0;
    λd,i,± := 0;
  end
end
```

2. Muster  $x$  durchgehen ( $x$  sei aus der Klasse  $k$ ):

```
for r = 1 to ‡ $\mathcal{DS}$  (durchlaufe Datensamples)
```

```
  if ( $p_i^k$  covers  $x$ ) then
```

3. Muster durch  $p_i^k$  überdeckt (cover):

$w_{i_*}^k := w_{i_*}^k + 1$ ; erhöhe nur einen einzigen Zähler  $i_*$  aus der Menge derjenigen  $i$ , für die  $p_i^k$   $x$  überdeckt  
adjustiere die  $\lambda_{d,i,\pm}^k$ , so dass sie  $x$  überdecken;

4. Neues Neuron einfügen (commit):

```
else
```

```
   $z_{m_k+1}^k := x$ ;
```

```
   $\lambda_{d,m_k+1,\pm} := 0$ ; Core-Bereich ist nur der Zentrumspunkt
```

```
   $\Lambda_{d,m_k+1,\pm}^k := \infty$ ; Support-Bereich ist unendlich
```

5.     **for**  $l \neq k, 1 \leq j \leq m_l$   
           Schrumpfe  $p_{m_k+1}^k$  anhand  $z_j^l$ , d.h.  
           shrink( $p_{m_k+1}^k, z_j^l$ );  
        **end**  
         $m_k := m_k + 1$ ;  
         $w_{m_k}^k := 1.0$ ;  
**end**

6. Konfliktneuronen verkleinern (shrink):

**for**  $l \neq k, 1 \leq j \leq m_l$   
       Schrumpfe  $p_j^l$  anhand  $x$ , d.h.  
       shrink( $p_j^l, x$ );  
   **end**  
**end**

Es folgt die Beschreibung des shrink-Moduls.

#### D.0.2 Algorithmus (Shrink-Mechanismus, Original):

Shrink-Algorithmus shrink( $p, x$ ) („minimaler Volumenverlust“):

$M := |z_{\tilde{d}} - x_{\tilde{d}}|$  mit  $\tilde{d} := \text{index} \left( \min_{1 \leq d \leq n} \left\{ \left| \frac{\Lambda_{d,\pm} - |z_d - x_d|}{\Lambda_{d,\pm}} \right| \right\} \right)$ ;

(Berechne  $\tilde{d}$  mittels +-Richtung in der Dimension  $d$  für  $\Lambda$ , falls  $z_d - x_d \geq 0$ ,  
sonst verwende die --Richtung, für alle *endlichen* Radian  $\Lambda$ .)

Falls  $M$  existiert, setze:

$\Lambda_{d,\text{min},\pm} := M$ ;

$\Lambda_{d,\text{bestfinite},\pm} := M$ , falls  $M \geq \sigma_{d,\text{min}}$ ;

$\Lambda_{d,\text{max},\pm} := \max\{|z_d - x_d| \mid 1 \leq d \leq n\}$ ;

(Berechne  $\Lambda_{d,\text{max},\pm}$  mittels +-Richtung in der Dimension  $d$  für  $\Lambda$ , falls  $z_d - x_d \geq 0$ ,  
sonst verwende die --Richtung, für alle *unendlichen* Radian  $\Lambda$ .)

Berechne ein neues  $\Lambda_{d,\pm}$ , d.h. ein *shrink* in einer Richtung für einen Radius:

**if**  $\Lambda_{d,\text{bestfinite},\pm}$  existiert **then**

$\Lambda_{d,\pm} := \Lambda_{d,\text{bestfinite},\pm}$ ;

**if**  $\Lambda_{d,\text{max},\pm} \geq \Lambda_{d,\text{min},\pm}$  **then**

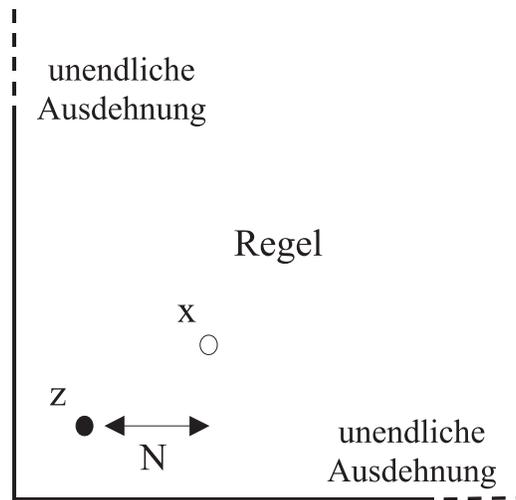
$\Lambda_{d,\pm} := \Lambda_{d,\text{max},\pm}$ ;

**else**

$\Lambda_{d,\pm} := \Lambda_{d,\text{min},\pm}$ ;

**end**

**end**



**Abbildung D.1:** Eine nicht dokumentierte Situation im Originalalgorithmus.

Anschließend wird noch kurz die Situation erläutert, die im Originalalgorithmus nicht beschrieben wurde, vgl. die Abb. D.1: Sei  $z$  ein Zentrum einer Regel und  $x$  das aktuelle Trainingssample, wobei  $x$  zu einer anderen Klasse als  $z$  gehöre, so dass geschrumpft werden muss. Liegt  $x$  von  $z$  aus gesehen bez. *allen* Dimensionen in einer Richtung, die eine unendliche Ausdehnung hat, so existiert das  $M$  aus dem Alg. D.0.1 nicht und damit auch nicht  $\Lambda_{d,\min,\pm}$  und  $\Lambda_{d,\text{bestfinite},\pm}$ , so dass die Abfrage „**if**  $\Lambda_{d,\max,\pm} \geq \Lambda_{d,\min,\pm}$  **then**“ nicht ausführbar ist. Deshalb wurde dieser Programmschritt verändert und eine Fallunterscheidung für  $N$  eingeführt, die darüber hinaus auch für unendliche Ausdehnungen den Parameter  $\sigma_{\min}$  berücksichtigt.



# Literaturverzeichnis

- [AB98] B. Arlt und R. Brause. The Principal Independent Components of Images. *Proc. of the Int. Conf. on Artificial Neural Networks (ICANN), Band 2*, 1998.
- [Adl90] K.-P. Adlassnig. Update on CADIAG-2: A Fuzzy Medical Expert System for General Internal Medicine. In Janko, W.H., Roubens, M. und Zimmermann, H.-J. (Hrsg.), *Progress in Fuzzy Sets and Systems*. Kluwer Academic Publishers. Seiten 1–6, 1990.
- [AHL98] A. Abu-Hanna und P. Lucas. Intelligent Prognostic Methods in Medicine. *Proc. of Computational Engineering in Systems Applications (CESA)*, Seiten 312–317, 1998. Internet.
- [AJBK00] H. Axer, J. Jantzen, G. Berks und D. Graf v. Keyserlink. Aphasia Classification Using Neural Networks. *Proc. of the 2000 European Symp. on Intelligent Techniques (ESIT)*, Seiten 111–115, 2000. Internet.
- [AS94] R. Agrawal und R. Skrikant. Fast Algorithms for Mining Association Rules. *Proc. of the 20<sup>th</sup> Int. Conf. on Very Large Databases (VLDB)*, Seiten 487–499, 1994. Internet.
- [AS95] R. Agrawal und R. Skrikant. Mining Sequential Patterns. *Proc. of the 11<sup>th</sup> Int. Conf. on Data Engineering (ICDE)*, Seiten 3–14, 1995. Internet.
- [Att98] H. Attias. Independent Factor Analysis. *Neural Computation*, 11:803–851, 1998. Internet.
- [AZ96] Pieter Adriaans und Rolf Zantige. *Data Mining*. Addison-Wesley, 1996.
- [B<sup>+</sup>92] R.C. Bone u.a. The ACCP/SCCM Concensus Conference. Definition for Sepsis and Organ Failure and Guidelines for the Use of the Innovative Therapies. *Chest*, 101:1644–1655, 1992.
- [B<sup>+</sup>96] K. Boegl u.a. Clinical Data Interpretation in MEDFRAME/CADIAG-4 Using Fuzzy Sets. *Biomedical Engineering-Applications, Basis & Communications*, 8:488–495, 1996.

- [B<sup>+</sup>97] K. Becker u.a. Design and Validation of an Intelligent Patient Monitoring and Alarm System Based on a Fuzzy Logic Process Model. *Artificial Intelligence in Medicine*, 11(1):33–53, 1997. Internet.
- [Bac96] R. Bacher. *Clusteranalyse*. Oldenbourg-Verlag, 2. Aufl., 1996.
- [Bau91] H. Bauer. *Wahrscheinlichkeitstheorie*. de Gruyter-Verlag, 4. völlig überarb. Aufl., 1991.
- [Bau96] H.-U. Bauer. Controlling the Magnification Factor of Self-Organizing Feature Maps. *Neural Computation*, 8:757–771, 1996.
- [BBMM98] E. Biganzoli, P. Boracchi, L. Mariani und E. Marubini. Feed Forward Neural Networks for the Analysis of Censored Survival Data: A Partial Logistic Regression Approach. *Statistics in Medicine*, 17:1169–1186, 1998.
- [BD95] M.R. Berthold und J. Diamond. Boosting the Performance of RBF Networks with Dynamic Decay Adjustment. *Proc. of Advances In Neural Information Processing Systems (NIPS)*, 7:521–528, 1995. Internet.
- [BD98] M.R. Berthold und J. Diamond. Constructive Training of Probabilistic Neural Networks. *Neurocomputing*, 19:167–183, 1998. Internet.
- [BDHH97] L. Bottaci, P.J. Drew, J.E. Hartley und M.B. Hadfield. Artificial Neural Networks Applied to Outcome Prediction for Colorectal Cancer Patients in Separate Institutions. *The Lancet*, 350:469–472, 1997.
- [Ber97] M.R. Berthold. *Konstruktives Training von Probabilistischen Neuronalen Netzen für die Musterklassifikation*. Infix-Verlag, 1997. Zugl. Dissertation Univ. Karlsruhe.
- [Ber99] M.R. Berthold. Fuzzy-Models and Potential Outliers. *Proc. of the 18<sup>th</sup> Int. Conf. of the North American Fuzzy Information Processing Soc. (NAFIPS)*, Seiten 532–535, 1999. Internet.
- [BF00] R. Brause und F. Friedrich. A Neuro-Fuzzy Approach as Medical Diagnostic Interface. *Proc. of the 9<sup>th</sup> European Symp. on Artificial Neural Networks (ESANN)*, Seiten 201–206, 2000. Internet.
- [BGI98] M. Bauer, U. Gather und M. Imhoff. Analysis of High Dimensional Data from Intensive Care Medicine. *Technical Report-Reihe des SFB 475*, Nr. 13/98, 1998. Internet.
- [BH95] M.R. Berthold und K.-P. Huber. From Radial to Rectangular Basis Functions: A New Approach for Rule Learning from Large Datasets. Internal Report 15-95, Univ. Karlsruhe, 1995. Internet.

- [BH98] M.R. Berthold und K.-P. Huber. Missing Values and Learning of Fuzzy Rules. *Int. Journ. of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(2):171–178, 1998. Internet.
- [BH99] M.R. Berthold und D.J. Hand (Hrsg.). *Intelligent Data Analysis: An Introduction*. Springer-Verlag, 1999.
- [BHP02] R. Brause, F. Hamker und J. Paetz. Septic Shock Diagnosis by Neural Networks and Rule Based Systems. In: M. Schmitt et al. (Hrsg.): *Computational Intelligence Processing in Medical Diagnosis*, Seiten 323–356. Springer-Verlag, 2002.
- [Bis95] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, 1995.
- [BKI00] C. Beierle und G. Kern-Isberner. *Methoden wissensbasierter Systeme*. Vieweg-Verlag, 2000.
- [Bla98] E. Blanzieri. *Learning Algorithms for Radial Basis Function Networks: Synthesis, Experiments and Cognitive Modelling*. Dissertation, Univ. and Polytechnic of Turin, Italien, 1998. Internet.
- [BLH99] R. Brause, T. Langsdorf und M. Hepp. Neural Data Mining for Credit Card Fraud Detection. *Proc. of the 11<sup>th</sup> IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI)*, Seiten 103–106, 1999. Ausführliche Version erschienen als Interner Bericht 7/99, FB Informatik, Univ. Frankfurt am Main, Internet.
- [BM95] I. Bratko und S. Muggleton. Applications of Inductive Logic Programming. *Communications of the ACM*, 38(11):65–70, 1995. Internet.
- [BM99] U. Bothner und F.W. Meissner. Wissen aus medizinischen Datenbanken nutzen. *Deutsches Ärzteblatt*, 95:A-1336–1338, 1999.
- [Boc73] H. H. Bock. *Automatische Klassifikation*. Vandenhoeck & Ruprecht, 1973.
- [Bol96] Toni Bollinger. Assoziationsregeln - Analyse eines Data Mining Verfahrens. *Informatik-Spektrum*, 19:257–261, 1996. Internet.
- [Bos92] K. Bosch. *Elementare Einführung in die angewandte Statistik*. Vieweg-Verlag, 4. durchges. Aufl., 1992.
- [BP01] C. Blake und W. Pratt. A Semantic Approach to Selecting Features from Text. *Proc. of the 1<sup>st</sup> Int. Conf. on Data Mining (ICDM)*, Seiten 59–66, 2001.

- [Bra95] R. Brause. *Neuronale Netze*. Teubner-Verlag, 2. überarb. und erw. Aufl., 1995.
- [Bra99] R. Brause. Revolutionieren Neuronale Netze unsere Vorhersagefähigkeiten? *Zentralblatt für Chirurgie*, 124:692–698, 1999. Internet.
- [Bru98] J. Bruske. Dynamische Zellstrukturen - Theorie eines KNN-Modells. Bericht Nr. 9809, Univ. Kiel. Zugl. Dissertation Univ. Kiel, 1998.
- [BSW86] P. Bauer, V. Scheiber und F.X. Wohlzogen. *Sequentielle statistische Verfahren*. Gustav Fischer Verlag, 1986.
- [BW97] A.D. Back und A.S. Weigend. A First Application of Independent Component Analysis to Extracting Structure from Stock Returns. *Int. Journ. of Neural Systems*, 8(5), 1997. Internet.
- [CF99] G. Castellano und A.M. Fanelli. Modeling Fuzzy Classification Systems with Compact Rule Base. *Proc. of the 1<sup>st</sup> Int. Conf. on Computational Intelligence for Modeling, Control and Automation (CIMCA)*, 1999. Internet.
- [Cha98] D. Charles. Constrained PCA Techniques for the Identification of Common Factors in Data. *Neurocomputing*, 22:145–156, 1998.
- [CM98] V. Cherkassky und F. Mulier. *Learning From Data: Concepts, Theory and Methods*. Wiley & Sons, 1998.
- [Com94] P. Comon. Independent Component Analysis, A New Concept? *Signal Processing*, 36:287–314, 1994.
- [CPn96] M.Á. Carreira-Perpiñán. A Review of Dimension Reduction Techniques. Technical Report, CS-96-09, Dept. of Computer Science, Univ. of Sheffield, Großbritannien, 1996. Internet.
- [CT91] T.M. Cover und J.A. Thomas. *Elements of Information Theory*. Wiley & Sons, 1991.
- [CZ00] G. Cheng und A. Zell. Externally Growing Cell Structures for Pattern Classification. *Proc. of the 2<sup>nd</sup> ICSC Symp. on Neural Computation (NC)*, Seiten 233–239, 2000. Internet.
- [DAG00] W. Duch, R. Adamczak und K. Grabczewski. A New Methodology of Extraction. Optimization and Application of Crisp and Fuzzy Logical Rules. *IEEE Transactions on Neural Networks*, 12(1):277–306, 2000. Internet.
- [Dap98] T. Dapper. *Dimensionsreduzierende Vorverarbeitungen für neuronale Netze*. Shaker-Verlag, 1998. Zugl. Dissertation Univ. Kassel 1997.

- [Dau98] M. Daumer. Online Monitoring von Changepoints: Erweiterungen und Alternativen zum Schwellenwertalarm. *Journ. für Anaesthesie und Intensivbehandlung*, 1:38–40, 1998. Internet.
- [DH73] R.O. Duda und P.E. Hart. *Pattern Recognition and Scene Analysis*. Wiley, 1973. Mittlerweile als 2. Aufl. erschienen: R.O. Duda, D.G. Stork und P.E. Hart, *Pattern Classification and Scene Analysis Part 1: Pattern Classification*, Wiley & Sons, 2000.
- [DL97] M. Dash und H. Liu. Feature Selection for Classification. *Intelligent Data Analysis*, 1:131–156, 1997. Internet.
- [DLK99] G. Dorffner, E. Leitgeb und H. Koller. A Comparison of Linear and Non-Linear Classifiers for the Detection of Coronary Artery Disease in Stress-ECG. *Proc. of the 7<sup>th</sup> Joint European Conf. on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM)*, Seiten 227–231, 1999.
- [DWS96] M. Denhard, A. Walter und C.-D. Schönwiese. Simulation globaler Klimaänderungen mit Hilfe neuronaler Netze. *Naturwissenschaftliche Rundschau*, 49(8), 1996.
- [DXGH] M.H. Dunham, Y. Xiao, L. Gruenwald und Z. Hossain. A Survey of Association Rules. Noch nicht veröffentlicht, Computer Science and Engineering Dept., Southern Methodist Univ., Dallas, Texas, USA, Internet.
- [E<sup>+</sup>97] A. Encke u.a. Evaluationsmodelle zur Therapieplanung / Standardisierung am Beispiel der Sepsis. *Langenbeck's Arch. Chir. Suppl. II (Kongreßbericht)*, Seiten 323–329, 1997.
- [F<sup>+</sup>97] A.M. Fein u.a. (Hrsg.). *Sepsis and Multiorgan Failure*. Williams & Wilkins, 1997.
- [FLF00] M.V. Fidelis, H.S. Lopes und A.A. Freitas. Discovering Comprehensible Classification Rules with a Genetic Algorithm. *Proc. of the Congress on Evolutionary Computation (CEC)*, Seiten 805–810, 2000. Internet.
- [FPB99] E.A. Fernandez, J. Presedo und S. Barro. An ECG Ischemic Detection System Based on Self-Organizing Maps and a Sigmoid Function Pre-processing Stage. *Proc. of the 7<sup>th</sup> Joint European Conf. on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM)*, Seiten 207–216, 1999.
- [Fri94] B. Fritzke. Fast Learning with Incremental RBF Networks. *Neural Processing Letters*, 1(1):2–5, 1994. Internet.

- [Fri97a] B. Fritzke. A Self-organizing Network That Can Follow Non-Stationary Distributions. *Proc. of the 7<sup>th</sup> Int. Conf. on Artificial Neural Networks (ICANN)*, Seiten 613–618, 1997. Internet.
- [Fri97b] B. Fritzke. Incremental Neuro-Fuzzy Systems. *Proc. SPIE's Optical Science, Engineering and Instrumentation: Applications of Fuzzy Logic Technology IV*, 3165:86–97, 1997. Internet.
- [Fri98] B. Fritzke. *Vektorbasierte Neuronale Netze*. Shaker-Verlag, 1998. Zugl. Habilitation Univ. Dresden, Internet.
- [Ger99] N.A. Gershenfeld. *The Nature of Mathematical Modeling*. Cambridge University Press, 1999.
- [GK01] V. Guralnik und G. Karypis. A Scalable Algorithm for Clustering Sequential Data. *Proc. of the 1<sup>st</sup> Int. Conf. on Data Mining (ICDM)*, Seiten 179–186, 2001.
- [Gol89] D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison Wesley, 1989.
- [GS97] J. Gamper und F. Steimann. Medizinische Expertensysteme - Eine kritische Betrachtung. In: *APIS-Zeitschrift für Politik, Ethik, Wissenschaft und Kultur im Gesundheitswesen*, 1997. Internet.
- [GU99] G. Guimarães und A. Ultsch. A Method for Temporal Knowledge Conversion. *Proc. of the 3<sup>rd</sup> Symp. on Intelligent Data Analysis (IDA)*, Seiten 369–380, 1999. Internet.
- [Gui98] G. Guimarães. *Eine Methode zur Entdeckung von komplexen Mustern in Zeitreihen mit Neuronalen Netzen und deren Überführung in eine symbolische Wissensrepräsentation*. Dissertation, Univ. Marburg, 1998. Internet.
- [H<sup>+</sup>00] M. Hantke u.a. Der SOFA-Score in der Beurteilung septischer Krankheitsbilder. *Der Chirurg*, 71:1270–1276, 2000. Internet.
- [Hag97] C. Hagen. *Neuronale Netze zur statistischen Datenanalyse*. Shaker-Verlag, 1997. Zugl. Dissertation TH Darmstadt.
- [Ham01] F. Hamker. Lifelong Learning Cell Structures – Continuously Learning Without Catastrophic Inference. *Neural Networks*, 14:551–573, 2001. Internet.
- [Har93] J. Hartung. *Statistik: Lehr- und Handbuch der angewandten Statistik*. R. Oldenbourg-Verlag, 9. durchges. Aufl., 1993.

- [Har98] V. Harms. *Biomathematik, Statistik und Dokumentation: Eine leichtverständliche Einführung*. Harms-Verlag, 7. überarb. Aufl., 1998.
- [Har00] R.M. Hardaway. A Review of Septic Shock. *American Surgeon*, 66(1):22–29, 2000. Internet.
- [Hay99] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall, 2. Aufl., 1999.
- [HB95] K.-P. Huber und M.R. Berthold. Building Precise Classifiers With Automatic Rule Extraction. *Proc. of the IEEE Int. Conf. on Neural Networks (ICNN)*, 3:1263–1268, 1995. Internet.
- [HB96] K.-P. Huber und M.R. Berthold. Analysis of Simulation Models with Fuzzy Graph Based Metamodeling. *Performance Evaluation*, 27/28:473–490, 1996. Internet.
- [HBBE98] E. Hanisch, M. Büssow, R. Brause und A. Encke. Individuelle Prognose bei kritisch kranken Patienten mit septischem Schock durch ein neuronales Netz? *Der Chirurg*, 69:77–81, 1998.
- [HC93] D.T. Hau und E.W. Coiera. Learning Qualitative Models of Dynamic Systems. *Machine Learning*, 26:177–211, 1993.
- [HCC92] J. Han, Y. Cai und N. Cercone. Knowledge Discovery in Databases: An Attribute-Oriented Approach. *Proc. of the 18<sup>th</sup> Int. Conf. on Very Large Databases (VLDB)*, Seiten 547–559, 1992. Internet.
- [HE95] J. Hartung und B. Elpelt. *Multivariate Statistik: Lehr- und Handbuch der angewandten Statistik*. R. Oldenbourg-Verlag, 1995.
- [HE01] E. Hanisch und A. Encke. Intensive Care Management in Abdominal Surgical Patients with Septic Complications. Erschienen in: E. Faist (Hrsg.) *Immunological Screening and Immunotherapy in Critically Ill Patients with Abdominal Infections*, Springer-Verlag, 2001.
- [Hel91] H. Helbig. *Künstliche Intelligenz und automatische Wissensverarbeitung*. Verlag Technik Berlin, 1991. Mittlerweile erschienen: 2. stark bearb. Aufl. 1996.
- [Heu95] H. Heuser. *Gewöhnliche Differentialgleichungen*. Teubner-Verlag, 3. durchges. Aufl., 1995.
- [HH97] T. Hellström und K. Holmström. Predicting the Stock Market. Technical Report IMA-TOM-1997-07, Dept. of Mathematics and Physics, Mälardalen Univ., Schweden, 1997. Internet.

- [HH98] D. Heinke und F. Hamker. Comparing Neural Networks: A Benchmark on Growing Neural Gas, Growing Cell Structures, and Fuzzy ARTMAP. *IEEE Transactions on Neural Networks*, 9(6):1279–1291, 1998.
- [HH99a] R.J. Hilderman und H. Hamilton. Knowledge Discovery and Interestingness Measures: A Survey. Technical Report CS 99-04, Dept. of Computer Science, Univ. of Regina, Kanada, 1999. Internet.
- [HH99b] R.J. Hilderman und H.J. Hamilton. Heuristic Measures of Interestingness. *Proc. of the 3<sup>rd</sup> European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Seiten 232–241, 1999.
- [HJ01a] R.J. Howlett und L.C. Jain (Hrsg.). *Radial Basis Function Networks 1: Recent Developments in Theory and Applications*. Springer-Verlag, 2001.
- [HJ01b] R.J. Howlett und L.C. Jain (Hrsg.). *Radial Basis Function Networks 2: New Advances in Design*. Springer-Verlag, 2001.
- [HK00] J. Han und M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufman Publishers, 2000.
- [HKK97] F. Höppner, F. Klawonn und R. Kruse. *Fuzzy-Clusteranalyse: Verfahren für die Bilderkennung, Klassifizierung und Datenanalyse*. Vieweg-Verlag, 1997.
- [HKLO96] L. Holmström, P. Koistinen, J. Laaksonen und E. Oja. Comparison of Neural and Statistical Classifiers - Theory and Practice. *Research Report A 13*, 1996. Rolf Nevanlinna Institute, Internet.
- [HKM<sup>+</sup>96] K. Hätönen, M. Klemettinen, H. Mannila, P. Ronkainen und H. Toivonen. Knowledge Discovery from Telecommunication Network Alarm Databases. *12<sup>th</sup> International Conference on Data Engineering (ICDE)*, Seiten 115–122, 1996. Internet.
- [HKO01] A. Hyvärinen, J. Karhunen und E. Oja. *Independent Component Analysis*. Wiley, 2001.
- [HM01] C.W. Hanson und B.E. Marshall. Artificial Intelligence Applications in the Intensive Care Unit. *Critical Care Medicine*, 29:427–435, 2001. Internet.
- [Hol93] J. Hollatz. *Integration von regelbasiertem Wissen in neuronale Netze*. Dissertation, TU München, 1993.
- [Höp01] F. Höppner. Discovery of Temporal Patterns – Learning Rules About The Qualitative Behaviour of Time Series. *Proc. of the 5<sup>th</sup> European*

- Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Seiten 192–203, 2001. Internet.
- [HPT<sup>+</sup>00] F. Hamker, J. Paetz, S. Thöne, R. Brause und E. Hanisch. Erkennung kritischer Zustände von Patienten mit der Diagnose „Septischer Schock“ mit einem RBF-Netz. Interner Bericht 4/00, Fachbereich Informatik, J.W. Goethe-Universität Frankfurt am Main, 2000. Internet.
- [HPY00] J. Han, J. Pei und Y. Yin. Mining Frequent Patterns Without Candidate Generation. *Proc. of the ACM SIGMOD Int. Conf. on Management of Data*, Seiten 1–12, 2000. Internet.
- [HS89] T. Hastie und W. Stuetzle. Principal Curves. *Journ. of the American Statistical Assoc.*, 84(406):502–516, 1989.
- [Hub98] K.-P. Huber. *Datenbasierte Metamodellierung mit automatisch erzeugten Fuzzy-Regeln*. VDI-Verlag, 1998. Fortschritt-Berichte / VDI: Reihe 10, Nr. 551. Zugl. Diss., Univ. Karlsruhe.
- [Hyv99] A. Hyvärinen. Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks*, 10(3):626–634, 1999. Internet.
- [J<sup>+</sup>00] T.-P. Jung u.a. Removing Electroencephalographic Artifacts by Blind Source Separation. *Psychophysiology*, 37:163–178, 2000. Internet.
- [Jol86] I.T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 1986.
- [JS93] J.-S.R. Jang und C.-T. Sun. Functional Equivalence Between Radial Basis Function Networks and Fuzzy Inference Systems. *IEEE Transactions on Neural Networks*, 4(1):156–159, 1993.
- [KDWZ85] W.A. Knaus, E. Draper, D.P. Wagner und J.E. Zimmerman. APACHE II: A Severity of Disease Classification System. *Critical Care Medicine*, 13(10):818–829, 1985.
- [KGK93] R. Kruse, J. Gerhardt und F. Klawonn. *Fuzzy-Systeme*. Teubner-Verlag, 1993.
- [Koh97] T. Kohonen. *Self-Organizing Maps*. Springer-Verlag, 2. Aufl., 1997. Mittlerweile erschienen: 3. Aufl. 2001.
- [Kra97] W. Krabs. *Mathematische Modellierung: eine Einführung in die Problematik*. Teubner-Verlag, 1997.
- [KS69] M.G. Kendall und A. Stuart. *The Advanced Theory of Statistics*, Band 1. Charles Griffin & Company Limited London, 3. Aufl., 1969. (einbändige 1. Aufl. 1958, zweibändige 1. Aufl. 1943).

- [Kui86] B. Kuipers. Qualitative Simulation. *Artificial Intelligence*, 29:289–338, 1986. Internet.
- [KWZ98] D. Krahl, U. Windheuser und F.-K. Zick. *Data Mining: Einsatz in der Praxis*. Addison-Wesley, 1998.
- [Lav99] N. Lavrač. Machine Learning for Data Mining in Medicine. *Proc. of the 7<sup>th</sup> Joint European Conf. on Artificial Intelligence in Medicine and Medical Decision Making (AIMDM)*, Seiten 57–60, 1999.
- [Lee98] T.-W. Lee. *Independent Component Analysis: Theory and Applications*. Kluwer Academic Publishers, 1998. Zugl. Dissertation FU Berlin.
- [Leh99] M. Lehtokangas. Modelling with Constructive Backpropagation. *Neural Networks*, 12:707–716, 1999.
- [LHM98] B. Liu, W. Hsu und Y. Ma. Integrating Classification and Association Rule Mining. *Proc. of the 4<sup>th</sup> Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, Seiten 80–86, 1998. Internet.
- [LKT95] S. Lemeshow, J. Klar und D. Teves. Outcome Prediction for Individual Intensive Care Patients: Useful, Misused, or Abused. *Intensive Care Medicine*, 21:770–776, 1995.
- [LTL<sup>+</sup>99] C.P. Lim, H.H. Toh, T.F. Lee, R.F. Harrison und R.L. Kennedy. Application of an Adaptive Neural Network to Medical Decision Support. *Int. Journ. of The Computer, The Internet and Management*, 7(1):9–20, 1999. Internet.
- [M<sup>+</sup>95] J.C. Marshall u.a. Multiple Organ Dysfunction Score: A Reliable Descriptor of a Complex Clinical Outcome. *Critical Care Medicine*, 23(10):1638–1652, 1995.
- [Mar92] T. Martinetz. *Selbstorganisierende neuronale Netzwerkmodelle zur Bewegungssteuerung*. Dissertation, TU München, 1992.
- [MBS93] T. Martinetz, S. Berkovich und K. Schulten. „Neural Gas“ Network for Vector Quantization and its Application to Time-Series Prediction. *IEEE Transactions on Neural Networks*, 4(4):558–569, 1993.
- [Mit97] T.M. Mitchell. *Machine Learning*. McGraw-Hill, 1997.
- [Mit99] T.M. Mitchell. Machine Learning and Data Mining. *Communications of the ACM*, 42(11):30–36, 1999. Internet.
- [MT95] I. Makoto und T. Takenobu. Cluster-Based Text Categorization: a Comparison of Category Search Strategies. Technical Report 95-TR0016. Dept. of Computer Science, Tokyo Institute of Technology., 1995. Internet.

- [Nau99] D. Nauck. Obtaining Interpretable Fuzzy Classification Rules from Medical Data. *Artificial Intelligence in Medicine*, 16(2):149–169, 1999. Internet.
- [Nau00a] D. Nauck. Adaptive Rule Weights in Neuro-Fuzzy Systems. *Neural Computing and Applications*, 9:60–70, 2000. Internet.
- [Nau00b] D. Nauck. Data Analysis with Neuro-Fuzzy Methods. Habilitationsschrift, Univ. Magdeburg, 2000. Internet.
- [NK97] D. Nauck und R. Kruse. A Neuro-Fuzzy Method to Learn Fuzzy Classification Rules from Data. *Fuzzy Sets and Systems*, 89:277–288, 1997.
- [NKK96] D. Nauck, F. Klawonn und R. Kruse. *Neuronale Netze und Fuzzy-Systeme*. Vieweg-Verlag, 2. überarb. und erw. Aufl., 1996.
- [NL96] E. Neugebauer und R. Lefering. Scoresysteme und Datenbanken in der Intensivmedizin - Notwendigkeit und Grenzen. *Intensivmedizin*, 33:445–447, 1996.
- [NRRS98] E. Neugebauer, D. Rixen, M. Raum und U. Schäfer. Thirty Years of Anti-Mediator Treatment in Sepsis and Septic Shock - What Have We Learned? *Langenbeck's Archives of Surgery*, 383(1):26–34, 1998.
- [O<sup>+</sup>01] C. Ordonez u.a. Mining Constrained Association Rules to Predict Heart Disease. *Proc. of the 1<sup>st</sup> Int. Conf. on Data Mining (ICDM)*, Seiten 433–440, 2001. Internet.
- [PAB] J. Paetz, B. Arlt und R. Brause. Rule Generation and Model Selection Used for Medical Diagnosis. Im Nov. 2001 übermittelt an: *Journ. of Intelligent and Fuzzy Systems*.
- [Pae] J. Paetz. Knowledge Based Approach to Septic Shock Patient Data Using a Neural Network with Trapezoidal Activation Functions. Im Aug. 2001 übermittelt an: *Artificial Intelligence in Medicine*.
- [Pae01] J. Paetz. Metric Rule Generation with Septic Shock Patient Data. *Proc. of the 1<sup>st</sup> Int. Conf. on Data Mining (ICDM), San Jose, USA*, Seiten 637–638, 2001.
- [Pae02a] J. Paetz. Durchschnittsbasierte Generalisierungsregeln Teil I: Grundlagen. *Frankfurter Informatik-Berichte Nr. 1/02*, Institut für Informatik, Fachbereich Biologie und Informatik, 2002.
- [Pae02b] J. Paetz. Some Remarks on Choosing a Method for Outcome Prediction. *Critical Care Medicine*, 30(3), 2002. Letter to Editor, erscheint im März 2002.

- [Pap91] A. Papoulis. *Probability, Random Variables and Stochastic Processes*. McGraw-Hill, 1991.
- [Pau95] N. Paul. *Medizinische Wissensbasen: Vom Wissensmodell zur Repräsentation*. Verlag Peter Lang, 1995. Zugl. Dissertation Univ. Münster.
- [Pau01] M. Pauen. Adaptive Analyse medizinischer Daten. Diplomarbeit, FB Informatik, J.W. Goethe-Universität Frankfurt, August 2001.
- [Paw82] Z. Pawlak. Rough Sets. *Int. Journ. of Computer and Information Sciences*, 11:341–356, 1982.
- [PB99] U. Pietruschka und R. Brause. Using Growing RBF-Nets in Rubber Industry Process Control. *Neural Computing and Applications*, 8(2):95–105, 1999. Internet.
- [PB01] J. Paetz und R. Brause. A Frequent Patterns Tree Approach for Rule Generation with Categorical Septic Shock Patient Data. *Proc. of the 2<sup>nd</sup> Int. Symp. on Medical Data Analysis (ISMDA)*, Seiten 207–212, 2001. Internet.
- [PB02] J. Paetz und R. Brause. Durchschnittsbasierte Generalisierungsregeln Teil II: Analyse von Daten septischer Schock-Patienten. Frankfurter Informatik-Berichte Nr. 2/02, Institut für Informatik, Fachbereich Biologie und Informatik, 2002.
- [PG98] A. Piras und A. Germond. Local Linear Correlation Analysis with the SOM. *Neurocomputing*, 21:79–90, 1998.
- [PHT00] J. Paetz, F. Hamker und S. Thöne. About the Analysis of Septic Shock Patient Data. *Proc. of the 1<sup>st</sup> Int. Symp. on Medical Data Analysis (ISMDA)*, Seiten 130–137, 2000. Internet.
- [PMK96] K. Pawelzik, K.-R. Müller und J. Kohlmorgen. Prediction of Mixtures. *6<sup>th</sup> Int. Conf. on Artificial Neural Networks (ICANN)*, Seiten 127–132, 1996.
- [Pre94] L. Prechelt. Proben1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules. Technical Report 21/94, 1994. Fakultät für Informatik, Univ. Karlsruhe.
- [Psc94] W. Pschyrembel. *Pschyrembel Klinisches Wörterbuch*. de Gruyter-Verlag, 257. neu bearb. Aufl., 1994. Erscheint regelmäßig in neuer Auflage.
- [PWZP00] C.-H. Perng, H. Wang, S.R. Zhang und D.S. Parker. Landmarks: A New Model for Similarity-Based Pattern Querying in Time Series Databases.

- 16<sup>th</sup> International Conference on Data Engineering (ICDE), Seiten 33–42, 2000. Internet.
- [Qui93] J.R. Quinlan. *C4.5: Programs For Machine Learning*. Morgan Kaufmann Publishers, 1993.
- [Rao73] C.R. Rao. *Lineare statistische Methoden und ihre Anwendungen*. Akademie-Verlag, 1973. Engl. Originalausgabe 1965.
- [Rap97] A. Rappelsberger. Neuronale Netze in der Medizin. Diplomarbeit, Institut für medizinische Computerwissenschaften, TU Wien, 1997. Internet.
- [RCE82] D.L. Reilly, L.N. Cooper und C. Elbaum. A Neural Model for Category Learning. *Biological Cybernetics*, 45:35–41, 1982.
- [RF94] J. Rogers und H.D. Fuller. Use of Daily Acute Physiology and Chronic Health Evaluation (APACHE) II Scores to Predict Individual Patient Survival Rate. *Critical Care Medicine*, 22:1402–1405, 1994.
- [Rit93] H. Ritter. Parametrized Self-Organizing Maps. *3<sup>rd</sup> Int. Conf. on Artificial Neural Networks (ICANN)*, Seiten 568–575, 1993.
- [RMS91] H. Ritter, T. Martinetz und K. Schulten. *Neuronale Netze: Eine Einführung in die Neuroinformatik selbstorganisierender Netzwerke*. Addison Wesley, 1991.
- [Roj93] R. Rojas. *Theorie der neuronalen Netze: Eine systematische Einführung*. Springer-Verlag, 1993.
- [Run00] T.A. Runkler. *Information Mining: Methoden, Algorithmen und Anwendungen intelligenter Datenanalyse*. Vieweg-Verlag, 2000.
- [SA96] R. Srikant und R. Agrawal. Mining Quantitative Association Rules in Large Relational Tables. *Proc. of the 1996 ACM SIGMOD Int. Conf. on Management of Data*, Seiten 1–12, 1996. Internet.
- [Sac92] L. Sachs. *Angewandte Statistik*. Springer-Verlag, 7. völlig neu bearb. Aufl., 1992.
- [San89] T.D. Sanger. Optimal Unsupervised Learning in a Single-Layer Linear Feedforward Neural Network. *Neural Networks*, 2:459–473, 1989.
- [SB99] R. Silipo und M.R. Berthold. Discriminative Power of Input Features in a Fuzzy Model. *Proc. of the 3<sup>rd</sup> Int. Symp. on Intelligent Data Analysis (IDA)*, Seiten 87–98, 1999. Internet.
- [Sch95] M. Schweizer. *Wissensanalyse und -erhebung mit Kohonen-Netzen am praktischen Beispiel der Lawinenprognose*. Dissertation, Univ. Zürich, 1995.

- [Sch97] B. Schölkopf. *Support Vector Learning*. Dissertation, TU Berlin, 1997. Internet.
- [Sch00] C. Schommer. *Konfirmative und explorative Synergiewirkungen im erkenntnisorientierten Informationszyklus von BAIK*. Dissertation, J.W. Goethe-Universität Frankfurt am Main, 2000.
- [Sch01] G. Schnitger. Algorithmisches Lernen. Vorlesungsskript SS 2001, FB Informatik, J.W. Goethe-Univ. Frankfurt am Main, 2001. Internet.
- [SE97] E. Schikuta und M. Erhart. The BANG-Clustering System: Grid-Based Data Analysis. *Proc. of the 2<sup>nd</sup> Int. Symp. on Intelligent Data Analysis (IDA)*, Seiten 513–524, 1997.
- [See97] H.-J. Seelos (Hrsg.). *Medizinische Informatik, Biometrie und Epidemiologie*. de Gruyter-Verlag, 1997.
- [SKRT00] M. Suistoma, A. Kari, E. Ruokonen und J. Takala. Sampling Rate Causes Bias in APACHE II and SAPS II Scores. *Intensive Care Medicine*, 26:1773–1778, 2000.
- [Spe90] D.F. Specht. Probabilistic Neural Networks. *Neural Networks*, 3:109–118, 1990.
- [SRV96] M. Schumacher, R. Roßner und W. Vach. Neural Networks and Logistic Regression: Part I. *Computational Statistics and Data Analysis*, 21:661–682, 1996. Teil 2 auf den Seiten 683–701.
- [SS95] R. Schlittgen und B.H.J. Streitberg. *Zeitreihenanalyse*. R. Oldenbourg-Verlag, 6. unwes. veränd. Aufl., 1995.
- [Ste98] F. Steimann. *Diagnostic Monitoring of Clinical Time Series*. Dissertation, TU Wien, 1998. Internet.
- [SVA97] R. Srikant, Q. Vu und R. Agrawal. Mining Association Rules with Item Constraints. *Proc. of the 3<sup>rd</sup> Int. Conf. on Knowledge Discovery in Databases and Data Mining (KDD)*, 1997. Internet.
- [SVS97] G. Schwarzer, W. Vach und W. Schumacher. On the Misuses of Artificial Neural Networks for Prognostic and Diagnostic Classification in Oncology. *FDM-Preprint-Reihe des Freiburger Zentrums für Datenanalyse und Modellbildung Nr. 46*, 1997.
- [SWR98] M.H. Schoenberg, M. Weiss und P. Radermacher. Outcome of Patients with Sepsis and Septic Shock after ICU Treatment. *Langenbeck's Archives of Surgery*, 383(1):44–48, 1998.

- [T<sup>+</sup>00] D. Towell u.a. Linear and Nonlinear Analysis of Hemodynamic Signals During Sepsis and Septic Shock. *Critical Care Medicine*, 28(6):2051–2057, 2000. Internet.
- [TG97] N. Tschichold-Gürman. The Neural Network Model RuleNet and its Application to Mobile Robot Navigation. *Fuzzy Sets and Systems*, 85(2):287–303, 1997. Internet.
- [The89] C.W. Therrien. *Decision Estimation and Classification: An Introduction to Pattern Recognition and Related Topics*. Wiley & Sons, 1989.
- [Thi98] F.M. Thiesing. *Analyse und Prognose von Zeitreihen mit Neuronalen Netzen*. Dissertation, Univ. Osnabrück, 1998.
- [THV94] S. Tan, J. Hao und J. Vandewalle. Identification of Nonlinear Systems by RBF Neural Networks. *Proc. of the 4<sup>th</sup> Int. Conf. on Artificial Neural Networks (ICANN)*, Seiten 1203–1206, 1994.
- [TLPD98] J. Timmer, M. Lauk, W. Pflieger und G. Deuschl. Cross-Spectral Analysis of Physiological Tremor and Muscle Activity, I: Theory and Application to Unsynchronised Electromyogram. *Biological Cybernetics*, 78:349–357, 1998. Teil II auf den Seiten 359–368.
- [TLSH91] L. Tong, R.-W. Liu, V.C. Soon und Y.-F. Huang. Indeterminacy and Identifiability of Blind Identification. *IEEE Transaction on Circuits and Systems*, 38(5):499–509, 1991.
- [Toi96] H. Toivonen. *Discovery of Frequent Patterns in Large Data Collections*. Dissertation, Univ. of Helsinki, 1996. Zugl. Series of Publications A, Report A-1996-5, Dept. of Computer Science, Univ. of Helsinki, Finland. Internet.
- [Ton90] H. Tong. *Non-Linear Time Series: A Dynamical System Approach*. Clarendon Press, 1990.
- [Tsu99a] S. Tsumoto. Knowledge Discovery in Medical Multi-Databases: A Rough Set Approach. *3<sup>rd</sup> European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Seiten 147–155, 1999.
- [Tsu99b] S. Tsumoto. Rule Discovery in Large Time-Series Medical Databases. *3<sup>rd</sup> European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, Seiten 23–31, 1999.
- [Tsu00a] H. Tsukimoto. Extracting Rules from Trained Neural Networks. *IEEE Transactions on Neural Networks*, 11(2):377–389, 2000. Internet.

- [Tsu00b] S. Tsumoto. Clinical Knowledge Discovery in Hospital Information Systems: Two Case Studies. *4<sup>th</sup> European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD)*, Seiten 652–656, 2000.
- [Übe71] K. Überla. *Faktorenanalyse*. Springer-Verlag, 1971.
- [UH91] A. Ultsch und K.-U. Höffgen. Automatische Wissensakquisition für Fuzzy-Expertensysteme aus selbstorganisierenden neuronalen Netzen. Forschungsbericht Nr.: 404, Univ. Dortmund, 1991.
- [UK97] K. Unertl und B.M. Kottler. Prognostische Scores in der Intensivmedizin. *Anaesthesist*, 46:471–480, 1997. Internet.
- [UKK95] A. Ultsch, D. Korus und T.O. Kleine. Integration of Neural Networks and Knowledge-Based Systems in Medicine. *Proc. of the 5<sup>th</sup>. Conf. on Artificial Intelligence in Medicine in Europe (AIME)*, Seiten 425–426, 1995.
- [Ult91] A. Ultsch. Konnektionistische Modelle und ihre Integration mit wissensbasierten Systemen. Forschungsbericht Nr.: 396, Univ. Dortmund, 1991. Zugl. Habilitationsschrift.
- [Unb97] R. Unbehauen. *Systemtheorie 1: Allgemeine Grundlagen, Signale und lineare Systeme im Zeit- und Frequenzbereich*. Oldenbourg-Verlag, 7. Aufl., 1997.
- [US90] A. Ultsch und H.P. Siemon. Kohonen’s Self-Organizing Feature Maps for Exploratory Data Analysis. *Proc. of the Int. Conf. on Neural Networks (ICNN)*, Seiten 305–308, 1990.
- [V<sup>+</sup>96] J.-L. Vincent u.a. The SOFA (Sepsis-Related Organ Failure Assessment) Score to Describe Organ Dysfunction/Failure. *Intensive Care Medicine*, 22:707–710, 1996.
- [V<sup>+</sup>98] J.-L. Vincent u.a. Use of the SOFA Score to Assess the Incidence of Organ Dysfunction/Failure in Intensive Care Units: Results of a Multicenter, Prospective Study. *Critical Care Medicine*, 26(11):1793–1800, 1998.
- [V<sup>+</sup>00] S. Vanderschueren u.a. Thrombocytopenia and Prognosis in Intensive Care. *Critical Care Medicine*, 28:1871–1876, 2000. Internet.
- [vBM97] J. H. van Bommel und M. A. Musen (Hrsg.). *Handbook of Medical Informatics*. Springer-Verlag, 1997.
- [Ves99] J. Vesanto. SOM-Based Data Visualization Methods. *Intelligent Data Analysis*, 3(2):111–126, 1999. Internet.

- [Vil96] T. Villmann. *Topologieerhaltung in selbstorganisierenden neuronalen Merkmalskarten*. Dissertation, Univ. Leipzig, 1996.
- [VWG00] T. Villmann, H. Wieland und M. Geyer. Data Mining and Knowledge Discovery in Medical Applications Using Self-Organizing Maps. *1<sup>st</sup> Int. Symp. on Medical Data Analysis (ISMDA)*, Seiten 138–151, 2000. Internet.
- [WBH98] S. Wade, M. Büssow und E. Hanisch. Epidemiologie von SIRS, Sepsis und septischem Schock bei chirurgischen Intensivpatienten. *Der Chirurg*, 69:648–655, 1998.
- [Whi90] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley & Sons, 1990.
- [Wic98] J.D. Wickel. Modellierung und Fuzzy-Regelung des menschlichen Blutzuckerhaushalts. Diplomarbeit, TU Darmstadt, FB Informatik, Januar 1998. Internet.
- [Wit00] T. Wittmann. *Wissensentdeckung in Datenbanken mit adaptiven Regelsystemen*. Verlag Peter Lang, 2000. Zugl. Dissertation Univ. Jena 1999.
- [WWT99] P.C. Wong, P. Whitney und J. Thomas. Visualizing Association Rules for Text Mining. *IEEE Symp. on Information Visualization (InfoVis)*, 1999. Internet.
- [YAC98] H.H. Yang, S.-I. Amari und A. Cichocki. Information-Theoretic Approach to Blind Source Separation of Sources in Nonlinear Mixture. *Signal Processing*, 64(3):291–300, 1998. Internet.
- [Zak98] M.J. Zaki. *Scalable Data Mining for Rules*. Dissertation, Univ. of Rochester, New York, USA, 1998. Internet.
- [ZCF<sup>+</sup>97] C. Zaniolo, S. Ceri, C. Faloutsos, R.T. Snodgrass, V.S. Subrahmanian und R. Zicari. *Advanced Database Systems*. Morgan Kaufmann Publishers, 1997.
- [Zel94] A. Zell. *Simulation neuronaler Netze*. Addison-Wesley, 1994. 1. unveränd. Nachdruck 1996.
- [Zim91] H.J. Zimmermann. *Fuzzy Set Theory and its Applications*. Kluwer Academic Publishers, 2. Aufl., 1991.



# Stichwortverzeichnis

- A-priori-Verfahren, 191
- Abdeckung, 58
- Abhängigkeitsanalyse, 35
- Abhängigkeitsgraph, 45
- Adaptives Verfahren, 3
- Ähnlichkeit von Regeln, 113, 144, 185, 239
- Äußerer Generalisierungsindex, 140, 188
- Aktivierungsfunktion, 15
- Alarm, 247
- Alarmhistorie, 248
- Alarmsystem, 14, 247
- Algorithmisches Lernen, 32
- ANALYST, 51
- Ansammlung, 118
- APACHE, 11
- APACHE1K-Datensatz, 206
- ARMA, 33
- ASK-Daten, 176, 269
- Assoziationsregel, 54, 142, 191
- Auswahlkriterien, 138
  
- Backpropagation, 17
- Bayes-Netz, 45
- Benchmark-Umgebung, 73, 118
- Biosignal, 36
- Black-Box-Verhalten, 47
- Blind Source Separation, 43
- Breast-Datensatz, 134
  
- Cancer1-Daten, 79, 95, 263
- CASA, 12
- Case-Based Reasoning, 146
- Changepoint, 14
- Cluster, 27
- Clusteranalyse, 27, 160, 187
- Concensus Conference, 147
  
- Core-Konflikt, 69
  
- Data Mining, 8, 10
- Datenanalyse, 2, 7, 10
- Datenvorverarbeitung, 150, 195
- Dendrogramm, 28, 187, 239
- Diabetes1-Daten, 82, 91, 263
- Differentialgleichung, 45
- Durchschnitt von Itemsets, 118
  
- Eigenschaften der durchschnittsbasier-  
ten Generalisierungsregeln, 121
- Entmischungsmatrix, 42
- Entscheidungsbaum, 60
- Evolutionärer Algorithmus, 61
- Existenz-Index, 142
  
- Faktorenanalyse, 40
- FastICA, 43
- Fehlende Werte, 90
- Fluch der Dimension, 35
- FP-Baum, 190
- Frühzeitigkeit, 107, 137, 166, 170, 176,  
238
- Fuzzy-Menge, 49
- Fuzzy-Niveau, 49, 93, 163, 167, 214
- Fuzzy-RecBF-DDA, 53, 291
- Fuzzy-RecBF-DDA, modifiziert, 66, 162,  
200, 242
- Fuzzy-WvRecBF-DDA, 108, 164
  
- Generalisierung nach dem Prinzip der  
Durchschnittsbildung, 120, 175,  
231
- Generalisierungsfähigkeit, 30
- Generalisierungsregel, 58, 142, 231
- Gewicht, 16

- Gewichtete Konfidenz, 123, 132  
 Gewichtete Konfidenz einer Regel, 58  
 Gewinner, 23  
 Glasgow Coma Score, 12  
 Gradientenverfahren, 17  
 Growing Cell Structures, 26  
 Growing Neural Gas, 26  
  
 Häufig12Vor-Datensatz, 161, 162  
 Häufig16K-Datensatz, 203, 214  
 Häufigkeit einer Regel, 55, 279  
 Hauptkomponentenanalyse, 38  
 Hauptzähler, 108  
 Hidden-Markov-Modell, 45  
 Hypothese, 12  
  
 ICA, 40, 83  
 Infektion, 148  
 Innerer Generalisierungsindex, 140, 188  
 IRIS-Daten, 77, 95, 134, 263  
 Item, 54  
 Itemset, 54  
 Itemvektor, 131  
  
 Klasse, 29  
 Klassifikation, 29, 132, 158, 176  
 Klassifikator auf der Basis der Generalisierungsregeln, 132, 189  
 Klinikdaten, 195, 268  
 Knowledge Discovery, 8  
 Kombination von Regelmengen, 61, 242  
 Kommerzielle Datenanalyse, 8  
 Konfidenz einer Regel, 55, 279  
 Kontextverwischung, 142  
 Korrelationsanalyse, 37, 153, 200  
  
 Länge eines Itemsets, 118  
 Leer-Index, 141  
 Lernrate, 16  
 Lernregel, 15  
 Leveleigenschaft, 122  
 Linguistische Variable, 50  
  
 Manuelle Regelgenerierung, 171  
 Markov-Modell, 45  
  
 Maschinelles Lernen, 8  
 Meannoise-Strategie, 90  
 MEDAN-Daten, 263  
 MEDAN-Projekt, 1  
 Mediannarrownoise-Strategie, 93  
 Merkmalswahl, 30  
 Modell, 44  
 Modifikator, 51  
 MODS, 12  
 MODS1K-Datensatz, 206  
 Multizentrenstudie, 13  
  
 Nachbarschaftsfunktion, 23  
 Nearby-Parameter, 86  
 Nebenzähler, 108  
 NEFCLASS, 53  
 Neuro-Fuzzy-System, 47  
 Neuronales Netz, 15  
 Nichttrivialer Durchschnitt von Itemsets, 118  
 No Free Lunch Theorem, 32  
 NormAusreißer-Daten, 110, 263  
  
 Optimale scharfe Regel, 101, 228  
 Overfitting, 30, 74  
  
 Patientenkurve, 9  
 Principal Component Analysis, 38  
 Proben-Daten, 263  
 Projektionsstrategie, 158  
  
 RBF-DDA-Netz, 19  
 RBF-Netz, 18  
 RecBF-DDA-Netz, 21  
 Regel, 22  
 Regelgenerierung, 46, 54, 161, 175, 200, 231  
 Regression, 37  
 Regressionsnetz, 38  
 Repräsentative Regelmenge, 114, 146, 228  
 ROC-Kurve, 10, 80, 83, 189, 207, 214, 232, 236, 273  
  
 SAPS II, 11

- SAPS1K-Datensatz, 206  
Satz von Comon, 42  
Scharfe Regel, 94  
Score, 11, 205  
Selbstorganisierende Karte, 22  
Sensitivität, 11  
Sensitivitätsanalyse, 38  
Sepsis, 148  
Septischer Schock, 148  
Shrink-Algorithmus, modifiziert, 68  
Signal, 36  
SIRS, 147  
Skalierung, 11  
SOFA, 11  
SOFA1K-Datensatz, 205, 218  
Software, 269  
Spezifität, 11  
Spiraldaten, 84, 99, 263  
Stochastische Unabhängigkeit, 42  
Studie, 13  
Subsysteme, 203, 266  
Support Vector Machine, 18  
SystDiasThrombo3K-Datensatz, 206, 225  
SystDiasThrombo3K/Sym35-Datensatz,  
244  
SystThrombo2K-Datensatz, 206, 219
- Testdaten, 31  
TISS-28, 12  
Trainingsdaten, 31  
Transinformation, 43
- U-Matrix, 24  
Überwachtes Lernen, 15  
Unüberwachtes Lernen, 15
- VC-Dimension, 32  
Vorstudie, 263  
Vorstudiendaten, 147, 263  
Vote-Datensatz, 133, 263
- Wichtig17Vor-Datensatz, 161, 167  
Wichtig6K-Datensatz, 205, 222  
Wichtigkeit, 103, 136, 166, 170, 178,  
215, 235  
Winner-Takes-All, 132  
Wissenschaftliche Datenanalyse, 8  
Zeitreihenprognose, 34  
Zentrum, 18  
Zugehörigkeitsfunktion, 49  
Zugehörigkeitsgrad, 49  
Zustand, 14  
Zustandsanalyse, 27, 158



# Lebenslauf

## Persönliche Daten

Name Jürgen Paetz  
Geburtsdatum und -ort 17. Oktober 1970 in Offenbach am Main

## Schulbildung

1977 - 1987 Grundsule, Förderstufe und Gymnasium  
in Obertshausen  
1987 - 1990 Gymnasium Mühlheim am Main  
Abschluss: Allgemeine Hochschulreife

## Berufsausbildung

Sept. 1990 - Febr. 1993 Ausbildung zum Mathematisch-technischen  
Assistenten (Informatiker) IHK  
bei der Telenorma GmbH (Bosch Telecom)  
in Frankfurt am Main

## Zivildienst

März 1993 - Mai 1994 Deutsches Rotes Kreuz in Frankfurt am Main

## Hochschulausbildung

Okt. 1992 - Okt. 1997 Studium der Mathematik mit Nebenfach  
Informatik an der FernUniversität Hagen  
(davon WS 92/93 - SS 94 als Teilzeitstudium)

Studienschwerpunkt: Funktionentheorie  
(Prof. Dr. A. Duma, Dr. G. Garske)  
Abschluss: Diplom-Mathematiker (sehr gut)

Diplomarbeit „Die Rungeschen Approximationssätze  
und die Hauptteil- sowie Nullstellenverteilungen“  
ausgezeichnet mit dem Preis der Gesellschaft der  
Freunde der FernUniversität für besondere Leistungen

weitere Studien umfassen u.a.:

Künstliche Intelligenz und Neuronale Netze  
(Prof. Dr. H. Helbig, Dr. A. Scherer)  
Software Engineering und Datenbanksysteme  
(Prof. Dr. H.-W. Six, Prof. Dr. G. Schlageter)  
Analysis, Funktionalanalysis  
(Prof. Dr. W. Beekmann, Prof. Dr. J. Boos)

Algebra (Dr. M. Schulte)  
Gewöhnliche und partielle Differentialgleichungen  
(Prof. Dr. A. Aleman, Prof. Dr. K. Veselić)  
Riemannsche Flächen  
(Prof. Dr. A. Duma, Prof. Dr. J. Wolfart)

seit Okt. 1998

Promotionsstudium im Fach Informatik an der  
Johann Wolfgang Goethe-Universität  
Frankfurt am Main  
Betreuer der Dissertation: Dr. habil. R. Brause

erschienene Publikationen: [PB01], [Pae01], [Pae02a],  
[PHT00], [HPT<sup>+</sup>00], [BHP02],  
[PB02]

im Druck: [Pae02b]

eingereicht: [Pae], [PAB]

### **Berufliche Tätigkeiten**

seit Sept. 1998

Wissenschaftlicher Mitarbeiter im DFG-Projekt  
„Medizinische Datenanalyse mit Neuronalen Netzen –  
Analyse des septischen Schocks“ am Klinikum der  
Johann Wolfgang Goethe-Universität Frankfurt am Main  
Projekt-Betreuer:  
Prof. Dr. Dr. med. E. Hanisch (FB Medizin) und  
PD Dr. habil. R. Brause (FB Biologie und Informatik)

Vorträge u.a. im Rahmen der MEDAN-Studententreffen  
am Klinikum der J.W. Goethe-Universität

Frankfurt am Main, im Februar 2002