Septic Shock Diagnosis by Neural Networks and Rule Based Systems

R. Brause[1], F. Hamker[2], J. Paetz[1]
[1]Department of Biology and Computer Science
J.W.Goethe-University, Frankfurt, Germany
{brause, paetz}@cs.uni-frankfurt.de
[2]California Institute of Technology, Pasadena, CA, USA
fred@klab.caltech.edu

Abstract In intensive care units physicians are aware of a high lethality rate of septic shock patients. In this contribution we present typical problems and results of a retrospective, data driven analysis based on two neural network methods applied on the data of two clinical studies.
Our approach includes necessary steps of data mining, i.e. building up a data base, cleaning and preprocessing the data and finally choosing an adequate analysis for the medical patient data. We chose two architectures based on supervised neural networks. The patient data is classified into two classes (survived and deceased) by a diagnosis based either on the black-box approach of a growing RBF network and otherwise on a second network which can be used to explain its diagnosis by human-understandable diagnostic rules. The advantages and drawbacks of these classification methods for an early warning system are discussed.

1   Introduction

In intensive care units (ICUs) there is one event which only rarely occurs but which indicates a very critical condition of the patient: the septic shock. For patients being in this condition the survival rate dramatically drops down to 40-50% which is not acceptable.
Up to now, there is neither a successful clinical therapy to deal with this problem nor are there reliable early warning criteria to avoid such a situation. The event of sepsis and septic shock is rare and therefore statistically not well represented. Due to this fact, neither physicians can

develop well grounded experience in this subject nor a statistical basis for this does exist. Therefore, the diagnosis of septic shock is still made too late, because at present there are no adequate tools to predict the progression of sepsis to septic shock. No diagnosis of septic shock can be made before organ dysfunction is manifest.

The criteria for abnormal inflammatory symptoms (systemic inflammatory response syndrome SIRS) are both non-specific and potentially restrictive [25]. Experience with the ACCP/SCCM Consensus Conference definitions in clinical trials has highlighted the fact that they are unable to accurately identify patients with septic shock who might respond to interventions targeted to bacterial infections and its consequences, identify patients at risk for septic shock and to improve the early diagnosis of septic shock.

Our main goal is the statement of diagnosis and treatment on the rational ground of septic data. By the data analysis we aim to

- help in guideline development by defining sufficient statistical criteria of SIRS, sepsis, and septic shock,
- provide the necessary prerequisites for a more successful conduct of innovative therapeutic approaches,
- give hints which variables are relevant for diagnosis and use them for further research,
- provide new approaches based on the statistical cause and context to sepsis diagnosis implementing cost-effective clinical practice guidelines for improved diagnosis and treatment of septic shock.

It should be underlined that our analysis does not provide medical evidence for the diagnostic rules and therapeutic guidelines obtained in the data mining process but facilitates the discovery of them. It is up to additional, rigorously controlled studies to verify the data mining proposals.

Instead, to assist physicians protecting patient's life, our main concern is not to make a final prognosis about the survival of the patients, but to build up an early warning system to give individual warnings about the patient's critical condition. The principle of such a system is shown in Figure 1.
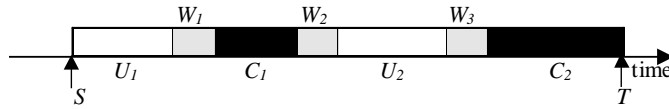
Figure 1 The concept of an early warning system. S = time of admission, T = time of death, shaded time intervals W_1, W_2, W_3: change of state, U_1, U_2 = uncritical period of time, C_1, C_2 = critical period of time

In clinical stay patients may change their state. Let us assume that in the periods of time U_i patients are uncritical, in C_j they are critical. Now, the aim of an early warning system is to give an alarm as early as possible in the transition phases W_k (k=1,3) and of course in C_j.

Critical illness states are defined as those states which are located in areas of the data showing a majority of measurements from deceased patients, see [16]. By detecting those states we expect to achieve a reliable warning, which should be as early as possible.

2 The Data

Very important for medical data analysis, especially for retrospective evaluations is the preprocessing of the data. In medical data mining, after data collection and problem definition, preprocessing is the third step. Clearly, the quality of the results from data analysis strongly depends on the successful execution of the previous steps. The three steps are an interdisciplinary work from data analysts and physicians and represent often the main work load.

In the following sections, we will show the main problems associated with our data. According to our experience, these problems are typical for medical data and should be taken into account in all approaches for medical data diagnosis. They include the selection of the number and kind of variables, treatment of small sets of mixed-case data with incorrect and missing values, selection of the subset of variables to analyze and the basic statistical proportions of the data.

2.1 The Data Context

Special care has to be taken in selecting and collecting patient data. In our case, the epidemiology of 656 intensive care unit patients (47 with a septic shock, 25 of them deceased) is elaborated in a study made be-

tween November 1995 and December 1997 at the clinic of the J.W.Goethe-University, Frankfurt am Main [36]. The data of this study and another study made in the same clinic between November 1993 and November 1995 is the basis of our work.

We set up a list of 140 variables, including readings (temperature, blood pressure, …), drugs (dobutrex, dobutamin, …) and therapy (diabetes, liver cirrhosis, …). Our data base consists of 874 patients. 70 patients of all had a septic shock. 27 of the septic shock patients and 69 of all the patients deceased.

2.2   Data Problems and Preprocessing

There are typical problems associated with medical data preprocessing. The problems and our approaches to maintain data quality are listed below.

- The **data set** is **too small** to produce reliable results. We tried to circumvent this problem by combining two different studies into one data pool.
- The medical **data** from the two **different studies** had to be **fused**. With the help of physicians we set up a common list of variables. Different units had to be adapted. Some variables are only measured in one of the two studies. It happened that time stamps were not clearly identifiable. Some data entries like *see above* or *zero* were not interpretable. So some database entries had to be ignored. The result is one common study with an unified relational database design including input and output programs and basic visualization programs.
- Naturally, our medical data material is very **inhomogeneous** (*case mix*), a fact that has to be emphasized. Each of the patients has a different period of time staying in the intensive care unit. For each patient a different number of variables (readings, drugs, therapies) was documented. So we had to select patients, variables and periods of time for the data base fusion. Because different data were measured at different times of day with different frequency (see Table 1), which gave hard to interpret multivariate time series, we used resampling methods to set the measurements in regular 24 hours time intervals.

Table 1 Averages of sampling intervals of four measured variables from all patients without any preprocessing. It is evident that a priori there is no state of the patient where all variables are measured synchronously.

variable	Average interval in [days : hours : min]
systolic blood pressure	1: 12: 11
temperature	1: 12: 31
thrombocytes	1: 18: 13
lactate	5: 0: 53

- **Typing errors** were detected by checking principal limit values of the variables. Blood pressure can not be 1200 (a missing decimal point). Typing errors in the date (03.12.96 instead of 30.12.96) were checked with the admission and the discharge day.

- A lot of variables showed a high number of **missing values** (internally coded with -9999) caused by faults or simply by seldom measurements, see Table 2.

Table 2 Available measurements of septic shock patients after 24-hours sampling for six variables

variable	measurements
systolic blood pressure	83.27 %
temperature	82.69 %
thrombocytes	73.60 %
inspiratorical O_2-concentration	65.81 %
lactate	18.38 %
lipase	1.45 %

The occurrence of faulty or missing values is a problem for many classical data analysis tools including some kinds of networks. The alternative of regularly sampled variables with a constant sample rate is not feasible in a medical environment. Since most of the samples are not necessary for the patient diagnosis or too expensive either in terms of unnecessary labor cost or in terms of high laboratory or device investment charges most of the important variables are measured only on demand in critical situations. Here, the sample rate depends also on the opinion of the supervising physician about

the patient's health conditions. Therefore, we have to live with the fact of missing values.

The treatment of missing values in the analysis with neural networks is described in more detail in section 3.

In conclusion, it is almost impossible to get 100% clean data from a medical data base of different patient records. Nevertheless, we have cleaned the data as good as possible with an enormous amount of time to allow analysis, see chapter 2.4.

For our task we heavily rely on the size of the data and their diagnostic quality. If the data contains too much inaccurate or missing entries we have no chance of building up a reliable early warning system even if it is principally possible.

2.3   Selecting Feature Variables

The data base contains about 140 variables of metric and categorical nature. For the small number of patients and samples we have, the number of variables is too high. Here, we encounter the important problem of "curse of dimensionality" [9] which is very hard to treat in the context of medical data acquisition. For a reliable classification the data space has to be sufficiently filled with data samples. If there is only a small number of samples available as in our case of septic shock patients, the training results become influenced by random: the classification boundaries depend on the values and sequence order of the samples.

An important approach to deal with this problem is the selection of a subset of "important" variables.

Which ones are important? There are systematic approaches for feature subset selection based on probabilities, see e.g. [21]. In our case, for analysis the physicians gave us recommendations which variables are the most important ones for a classification, based on their experience. The chosen variable set V is composed of n=16 variables: pO_2 (arterial) [mmHg], pCO_2 (arterial) [mmHg], pH, leukocytes [1000/µl], thrombo-plastin time (TPZ) [%], thrombocytes [1000/µl], lactate [mg/dl], creatinin [mg/dl], heart frequency [1/min], volume of urine [ml/24h], systolic blood pressure [mmHg], frequency of artificial respiratory [1/min], in-

spiratorical O$_2$-concentration [%], medication with antithrombine III AT3 [%], medication with dopamine and dobutrex [μg/(kg·min)].

2.4   Basic Statistical Analysis

Now, we give an impression of the basic statistical properties for our data set. We are aware of the problem that a relative small data set of subjects (in our case only 70 patients) with a septic shock, including missing values in some variables, are not sufficient for excellent results but we can give some hints and first results in the right direction based on the available data.

For the basic statistics, we calculated some statistical standard measures for each of the variables (mean, standard deviation etc.) including all patients or only the septic shock patients combined with all days or comprising only the last day of their stay in the intensive care unit.

Q-Q-plots show that the distributions are usually normal with an huge overlap of values from deceased and survived patients; the pure probability distributions do not show any significant difference. Figure 2 shows two histograms for two variables.
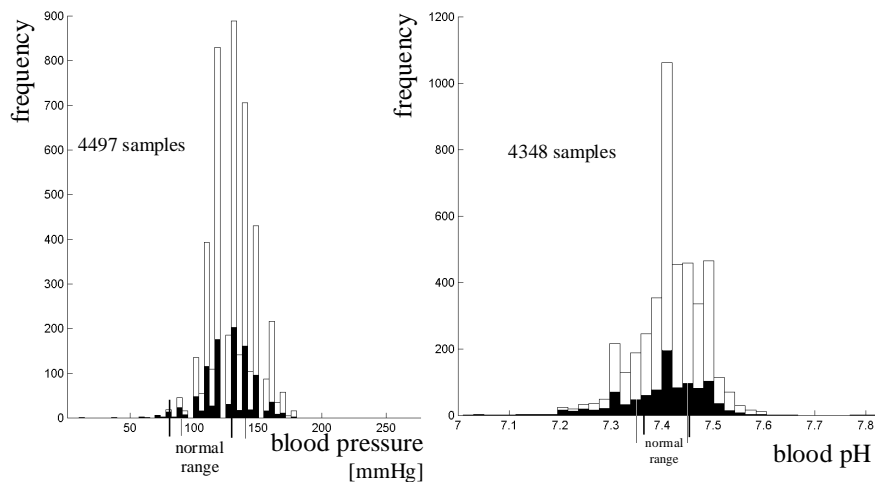

Figure 2 Histograms for a) systolic blood pressure and b) pH value for survived (white boxes) and deceased patients (black boxes). Clearly, the huge overlap of the two sample classes makes a classification very difficult.

If some variable values are correlated, it will not show up in the distributional plots. So, we checked this case also. A correlation analysis of the data shows high absolute values for the correlations between medicaments and variables, so surely the medicaments complicate the data analysis. Correlations between variables and {survived, deceased} are not high or not significant.

More interesting are the correlations COR(X,Z) calculated one time with the sets X_d, Z_d of samples from deceased and one time with the sets X_s, Z_s of samples from survived patients. The corresponding differences taken from all patients and all days is listed in Table 3. The significance level was calculated with SPSS 9.0. The correlations with significance level 0.01 are printed in bold font.

Table 3 Correlations between two variables (all patients, all days of hospital stay) with the highest correlation differences ≥ 0.3 between survived and deceased patients and frequency of measurement of each variable $\geq 20\%$. Significant correlations (level 0.01) are printed in bold letters. GGT is the abbreviation of gammaglutamyltransferase.

variable X	variable Z	COR(X_s,Z_s)	COR(X_d,Z_d)	diff.
inspir. O_2-concentration	pH	-0.03	**-0.39**	0.36
leukocytes	GGT	0.00	**0.32**	0.32
iron (Fe)	GGT	**0.31**	0.01	0.30
(total) bilirubin	urea	**0.26**	-0.07	0.33
urea	creatinin	**0.14**	**0.57**	0.43
fibrinogen	creatinin in urine	0.05	**-0.31**	0.36
arterial pO_2	potassium(K)	**-0.13**	**0.18**	0.31
thromboplastin time TPZ	chloride	**0.24**	-0.07	0.31

Both correlation values for the pairs urea, creatinin and arterial pO_2, potassium are significant (level 0.01), so that the difference could be an indicator for survived or deceased patients. Therefore, these variables should be measured very often to calculate the correlation in a time window during the patients actual stay at hospital. If they turn out to be too high, early warnings could be triggered.

Also, by training a neural network with the correlation values one can find out the exact threshold for a warning based on correlation values or combinations or modifications of such values (for first results see [16]).

Generally, this result seems to be reasonable because physicians reported that the interdependence of variables, measured from critical illness patients, could be disturbed by septic shock [34].

3 The Neural Network Approach to Diagnosis

In the last years many authors contributed to machine learning, data mining, intelligent data analysis and neural networks in medicine (see e.g. [4][23] and [5]). For our problem of septic shock diagnosis supervised neural networks have the advantages of nonlinear classification, fault tolerance for missing data, learning from data and generalization ability. The aim of our contribution is not a comparison of statistical methods with neural network results (e.g. see [31]) but to select an appropriate method that can be adapted to our data. Here, our aim is to detect critical illness states with a classification method.

It is widely accepted in the medical community that the septic shock dynamics are strictly nonlinear [34][32]. After preliminary tests we also concluded that linear classifiers are not suitable for classification in this case. In addition, most nonlinear classification methods also detect linear separability if it exists.

3.1 The Network

The neural network chosen for our classification task is a modified version of the supervised growing neural gas (abbr. SGNG, see [12][13][8])[1]. Compared to the classical multilayer perceptron trained with backpropagation (see [18]) which has reached a wide public, this network achieved similar results on classification tasks, see [19]. The results are presented in section 3.4.

The algorithm with our improvements and its parameters is noted in detail in [16]. It is based on the idea of radial basis functions (abbr. RBF, see [18]). The centers of the radial basis functions are connected through an additional graph that is adapted within the learning process, see appendix A. The graph structure allows to adapt not only the pa-

---

[1] Logistic regression is a statistical alternative to supervised neural networks

rameters (weights, radii) of the best matching neuron but also those of its neighbors (adjacent neurons). Its additional advantage is the ability to insert neurons within the learning process to adapt its structure to the data, see appendix A.

3.1.1 The Network Architecture

The neural network is build by two layers: the hidden layer (representation layer) and the output layer which indicates the classification decision for an input pattern.

The cell structure of the representation layer forms a parametrical graph P=P(G,S) where each node $v_i \in$ V (each neuron) has just one weight vector $w_i \in$ S with S $\subset \mathbb{R}^n$. The neighborhood relations between the nodes are defined by a non-directional graph G (see [24][7]) were G=G(V,E) consists of a set of nodes V=$\{v_1,...,v_m\}$ and a set of edges E=$\{e_1,...,e_m\}$. An incidence function f maps each edge to an unordered pair $[v_i, v_j]$ of nodes v_i, v_j, the end points or end nodes. The neighbors of a node are defined as those nodes which share an edge with it. For the graph G=G(V,E) the set N_i of neighbors of node i is defined by the equation

$$N_i = \{ v_j \: / \: \exists \: e_k \: : \: f(\, e_k \,) = [v_i,\; v_j] \: \}.$$ (1)

Each node of the representation layer computes its activity y_i by the RBF activation function

$$y_i = e^{-\frac{\|\mathbf{x}-\mathbf{w}_i\|^2}{\sigma_i^2}} \quad \forall \; v_i \in \text{G}$$ (2)

where the width of the Gaussian function, the standard deviation σ_i, is given by the mean edge length s_i of all edges connected to node v_i.

The m output neurons representing m classes are linear, i.e. their activity is computed as

$$z_j = \sum_{v_i \in \text{G}} w_{ji}^{\text{out}} \cdot y_i \quad \forall \; v_i \in \text{G}$$ (3)

using the output layer weight vectors $\mathbf{w}_j^{\text{out}} = (w_{j1}^{\text{out}},...,w_{jn}^{\text{out}})$. The decision for class k is based on the maximal output activity by a winner-takes-all mechanism.

$$C_k = \max_j (z_j + \theta_j)$$ (4)

which is influenced by a sensitivity parameter θ_j.

3.1.2   Treatment of missing values

Networks like the Supervised Growing Neural Gas (SGNG) present an alternative to dropping samples where only a few number of values are absent. By learning also with a fewer number of values more samples can be used for training and testing.

To achieve knowledge about a patient being in a critical illness condition, we need to classify the vectors $\mathbf{x}=(x_1,...,x_n)^t$ composed of measurements or drugs x_i, $i=1,...,n$ with the outcome y_s (survived) resp. y_d (deceased). For the n-dimensional data vector $\mathbf{x}$, we projected the vector $\mathbf{x}$ such that no missing value is in the projected vector $\mathbf{x}_p := (x_{i_1},...,x_{i_m})^t$, $\{i_1,...,i_m\} \subset \{1,...,n\}$, $m \leq n$, $x_{i_1},...,x_{i_m}$ are not missing values. Due to the fact that the SGNG is based only on distance calculations between vectors, it is possible to apply this standard projection argument to the adaptation and activation calculations of the SGNG, so that all calculations are done with the projected vectors $\mathbf{x}_p$. To find the best matching neuron we compute the Euclidean distance d_i by

$$d_i = \frac{1}{|I|}\sqrt{\sum_{l \in I}(x_l - w_{il})^2} \ , \quad I = \{l \mid x_l \text{ exists}\} \qquad (5)$$

Here, we take only the existing values, excluding explicitly the missing ones. The computation of the activity y_i in eq.(2) is done in the same way.

Certainly, there is a probable error involved in the classification when not all values are present, depending on the data set. Preliminary experiments showed that in our case it is not appropriate to project to less than half the variables. Therefore we used only samples containing more than 50% valid variables. This procedure causes a statistical bias, but we believe that it is not high because the most part of the data is missing randomly.

3.2   Training and Diagnosis

It is well known that the training performance of learning systems often does not reflect the performance on unknown data. This is due the fact that the system often adapts well on training to the particularities of the

training data. In the worst case a network just stores the training pattern and acts as an associative memory.

3.2.1 The training and test performance

In order to test the real generalization abilities of a network to unknown data, it must be tested by classified unknown data, the *test data*. As we already mentioned in section 2.3, the numbers of patients and samples are not very high in most medical applications. Therefore, the classical scheme of dividing all available data into training and test data is not possible, because the bigger we choose the training data set the smaller the test data set will be and the test results become vague. Choosing a small training set does no good either, because the trained state becomes also arbitrary, depending on the particularities of the training set composition. Here, special strategies are necessary.

One of the most used methods is the *p-fold cross validation* [37] [14]. Here, the whole data set is divided into *p* parts of equal size. The training is done in cycles or epochs where in each epoch one part (subset) of the data set is chosen as test set and the remaining *p-1* parts of the data are used for training. This can be done *p* times. The test performance is computed as the mean value of all *p* epoch tests.

The concept can be extended to use all *M* samples as parts such that the test is done by just one sample. This is known as the *leave-one-out* method [26] and was used in our report [16]. It corresponds to the situation of an *online learning* early warning system trained on a set of patients and asked for the diagnosis for a new arriving patient.

For the results of this paper, we did not use this but simply divided the samples into 75% training and 25% test patterns.

3.2.2 The problem of medical data partition

There is another problem, especially for training with medical data. We might not distinguish between the data of different patients, treat all samples equal and partition the data set of labeled samples randomly. Thus, data from the same patient appears both in the training and in the test set.
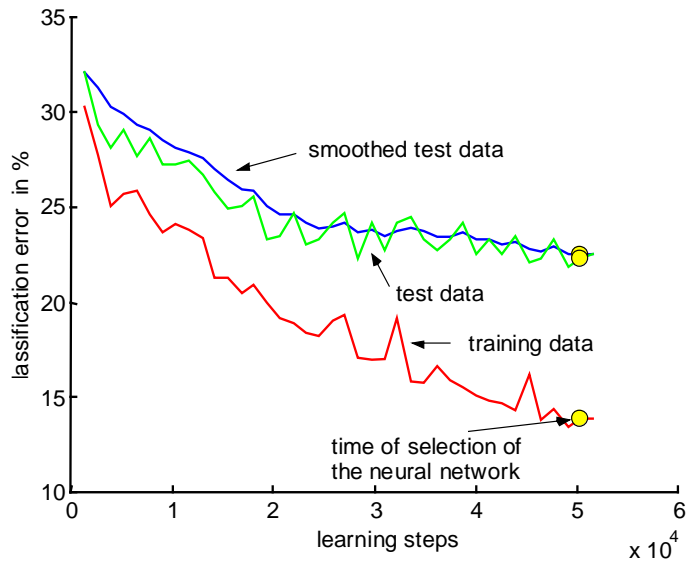
Figure 3 Random division of the data by samples

This is shown in Figure 3. In contrast to this, the parts can be chosen such that all samples of one patient are either only in the training set or in the test set. The resulting performance is shown in Figure 4
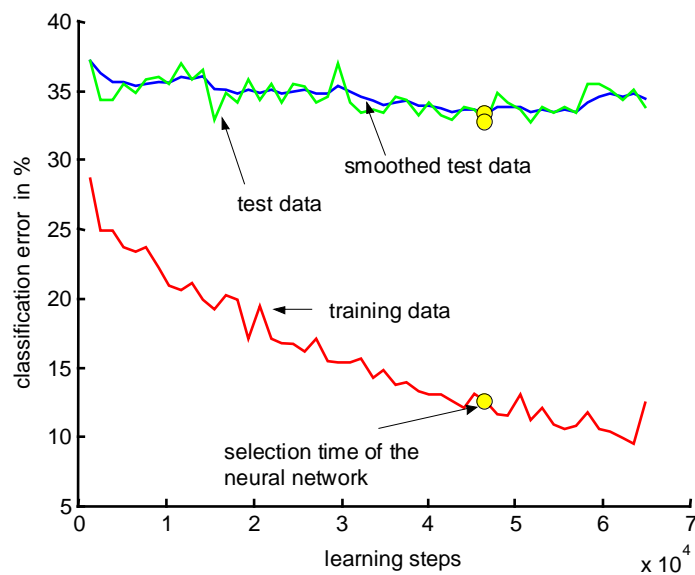

Figure 4 Division of the data by patients

335

It turns out that the result with the random partition of samples is much better. But does this result reflect the usage reality of an early warning system? By choosing the random partition, we assume that an early warning system already knows several samples of a patient from the training. This assumption is generally not true in clinical usage.

We have to face the fact that patient data is very individual and it is difficult to generalize from one patient to another. Ignoring this fact would pretend better results than a real system could practically achieve.

3.3   Selection and Validation of a Neural Network

One of the important parameters to get a non-overtrained, generalizing network is the time when the training has to be stopped. This time step is obtained by watching the performance of the net on the test set during training. First, the test error decreases in the adaptation process. When the test error increases again, the training should be stopped. Since the samples are randomized, the error should be smoothed in order to be approximately precise. This is shown in Figure 3 at the small circles.

There are three main approaches for selecting a suitable grown network by cross validation :

a) The test set is quite good, but choosing a network by the test set performance makes the choice depend on test set peculiarities. To avoid this, we might choose a third set of independent samples, the validation set. For instance, we might use 50% of the samples for training, 25% for testing and 25% for validation. In the medical environment where we have only a small number of patients and a small number of hand-coded variables, the advantage of independent test and validation becomes obsolete due to the random properties of the very small test and validation sets. The sets differ heavily in their proportions and are no more representative, the stopping criterion and the performance prediction becomes very arbitrary. This can be observed by a high deviation of the performance mean in the *p-fold* cross validation process.

b) The second approach uses the test set both as stopping criterion (choice of the appropriate network) and for validation, i.e. prediction. This improves the performance on the test set, but decreases the prediction performance on unknown data compared to an additional independent validation set. Nevertheless, since we are able to use more of our samples for training, the result becomes closer to the result a real application could achieve.

c) To achieve a maximal training performance in the presence of only a very small number of samples we might use all the samples for training and estimate the best stopping point by the training performance development alone without any explicit test. This includes subjective estimation and does not avoid random deviations of a good state.

The peculiarities of the choice for the sets can be decreased by smoothing the performance results. This can be obtained by taking the moving average instead of the raw value.

In our case we had only 70 patients with the diagnosis "septic shock". The high individual difference between the patients did not encourage us to choose different test and validation sets. Here we chose a test set that contains about 25% of the samples and ensured that all samples in the test set are from patients which are not used in the training set. In another investigation [16], we choose the leave-one-patient-out method to increase the size of the training set and to check each patient under the assumption that all other patients are known.

How reliable is such a diagnostic statement? In classical regression analysis, confidence intervals are given. In cases where there is no probability distribution information available as in our case this is very hard to do, see [17]. There are some attempts to introduce confidence intervals in neural networks [10][22][33], but with moderate success. Therefore, we decided to vary the context of testing as much as possible and give as result the deviation, maximum and minimum values additionally to the mean performance.

For the individual case the activity of the classification node of the second layer may be taken as an performance measure for the individual diagnosis ([16]).

3.4　Results for septic shock diagnosis

Our classification is based on 2068 measurement vectors (16-dimensional samples) from variable set *V* taken from 70 septic shock patients. 348 samples were deleted because of too many missing values within the sample. With 75% of the 1720 remaining samples the SGNG was trained and with 25% samples from completely other patients than in the training set it was tested.

The variables were normalized (mean 0, standard deviation 1) for analysis.

The network chosen was the one with the lowest error on the smoothed test error function. Three repetitions of the complete learning process with different, randomly selected divisions of the data were made. The results are presented in Table 4.

Table 4 Correct classifications, sensitivity, specificity with standard deviation, minimum and maximum in % from three repetitions.

measure	mean value	standard deviation	minimum	maximum
correct classi-fication	67.84	6.96	61.17	75.05
sensitivity	24.94	4.85	19.38	28.30
specificity	91.61	2.53	89.74	94.49

To achieve a generally applicable result ten repetitions would be better, but here it is already clear: with the low number of data samples the results can only have prototypical character, even with more cleverly devised benchmark strategies. Some additional results are reported in [16]. On average we have an alarm rate (= 1 − specificity) of 8.39% for survived patients showing also a critical state and a detection of about 1 out of 4 critical illness states. For such a complex problem it is a not too bad, but clearly no excellent result. An explanation for this low number is grounded in the different, individual measurements of each patient. To give an impression of the warnings over time we show in Figure 5 the resulting warnings from classification for 7 out of 24 deceased patients with septic shock.
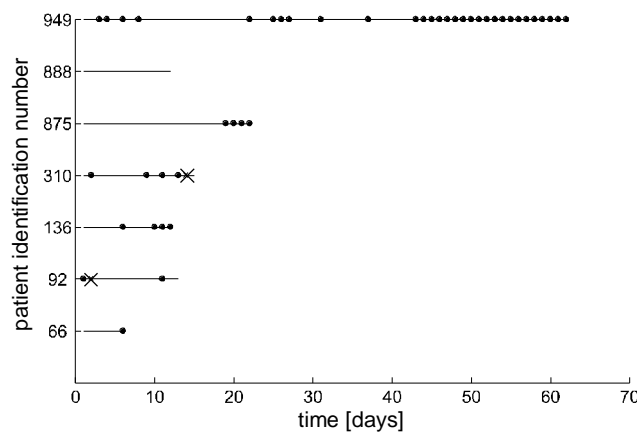
Figure 5 Deceased septic shock patients during their hospital stay with warnings (dot markers). A too high number of missing values causes some missing states (crosses). If there is no marker then no warning will be given.

Not for each deceased patient exists a warning (patient with number 888) and some warnings are given too late (patient with number 66), i.e. the physicians knew already that the patient had become critical. So the ideal time to warn the physician has not yet been found for all patients and remains as future work.

4 The Neuro-Fuzzy Approach to Rule Generation

Results of classification procedures could provide a helpful tool for medical diagnosis. Nevertheless, in practice physicians are highly trained and skilled people who do not accept the diagnosis of an unknown machine (black box) in their routine. For real applications, the diagnosis machine should be become transparent, i.e. the diagnosis should explain the reasons for classification. Whereas the explanation component is obvious in classical symbolic expert system tools, neural network tools hardly explain their decisions. This is also true for the SGNG network used in the previous section.

Therefore, as important alternative in this section we consider a classification by learning classification rules which can be inspected by the physician. Actual approaches to rule generation consider supervised learning neuro-fuzzy-methods [20][14], especially for medical applications [6][27].

Usually, medical data contain both metric and categorical variables. Here, our data is substantially based on *metric* variables, so in the following we consider the process of rule generation only for metric variables.

We devised an algorithm based on rectangular basis functions for the rule generation approach for metric variables which we apply to the septic shock patient data.

4.1   The rule extraction network

First we describe the fundamental ideas of the algorithm and then we give a detailed description of it. The network structure – as we use it for two classes – is shown in Figure 6.
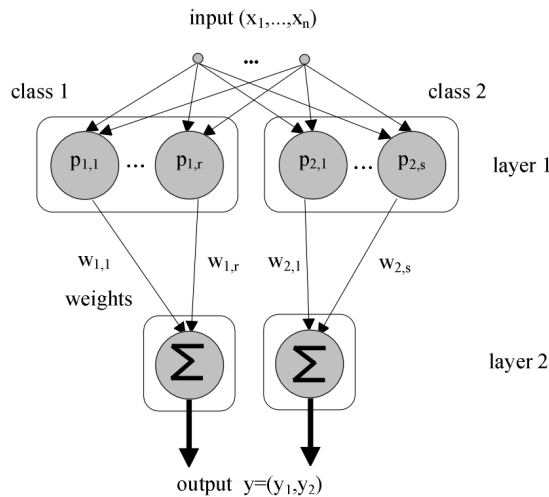

Figure 6   Network structure for two classes. Each class in
layer 1 has its individual number of neurons.

The 2-layer network has neurons - separately for every class - in layer 1. The r neurons $p_{1,1}$, ..., $p_{1,r}$ belong to class 1 and the s neurons $p_{2,1}$, ..., $p_{2,s}$ to class 2. The activation functions of the neurons represent rule prototypes using different asymmetrical trapezoidal fuzzy activation functions $R_{1,1}$, ..., $R_{1,r}$ and $R_{2,1}$, ..., $R_{2,s}$ with image [0,1].

   The algorithm is an improved version of the RecBFN algorithm of Huber and Berthold [20] which in turn is based on radial basis functions [18] with dynamic decay adjustment [2][3]. During the **learning phase** the input data is passed unmodified to layer 1. Then all neurons are adapted, i.e. the sides of the smaller rectangles (= *core rules*) and the sides of the larger rectangles (= *support rules*) of the fuzzy activation function graph are adapted to the data samples, see Figure 7.
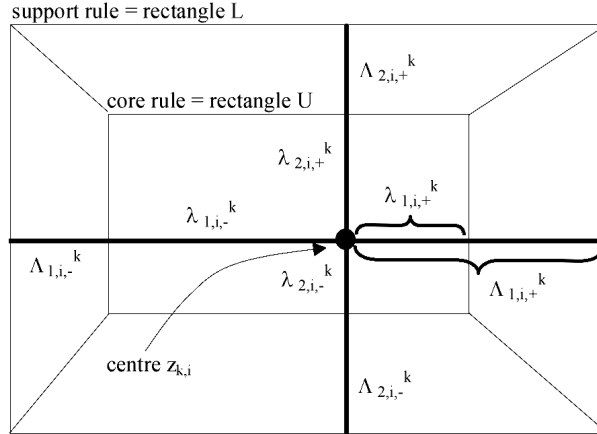
Figure 7 Two-dimensional projection (bird's view) of the trapezoidal function of one neuron with support and core rule and parameters of the algorithm in appendix B and C, representing one fuzzy rule for class k (see Figure 1 in [20]). U is the upper and L the lower rectangle of the trapezoid.

This happens in four phases for every new training data sample vector $\mathbf{x} \in \mathbb{R}^n$ of class k with n as dimension of the data space,

(1) *cover*: if $\mathbf{x}$ lies in the region of the support rule for all neurons – generated so far – of the same class k as $\mathbf{x}$, expand one side of the core rule to cover $\mathbf{x}$ and increment the weight of the neuron.

(2) *commit*: if no support rule covers $\mathbf{x}$, insert a new neuron p with center $\mathbf{x}$ of the same class k and set its weight to one; the expansions of the sides are initially set to infinite.

(3) *shrink committed neuron*: for a committed neuron shrink the volume of the support and the core rectangle within *one* dimension of the neuron in dependency of the neurons belonging to other classes.

(4) *shrink conflict neurons*: for all neurons, belonging to another class not equal to k, shrink the volume of both rectangles within *one* dimension in dependency of $\mathbf{x}$.

For details of the main algorithm and the shrinking procedure see appendix B and C.

An advantage of the method is its simplicity that softens the combinatorial explosion in rule generation by its cover-commit-shrink-procedure. By side expansions of the fuzzy activation function to infinite it is possible to find out the variables that are not interesting for a rule, see rules

(9) and (10) below. It is also directly possible to integrate a-priori known rules after fuzzification.

Finally, **classification activity** is done by a *winner-takes-all* mechanism, i.e. the calculation of the output $y_k = y_k(\mathbf{x})$ as the sum of the weights multiplied by fuzzy activation for every class $k \in \{1, 2\}$:

$$y_1 := w_{1,1} \cdot R_{1,1} + \dots + w_{1,r} \cdot R_{1,r} \qquad (6)$$
$$y_2 := w_{2,1} \cdot R_{2,1} + \dots + w_{2,s} \cdot R_{2,s} \qquad (7)$$

Then, choose class c_{max} as classification result, where c_{max} is the class label of the maximal output:

$$c_{max} := \text{class} \left(\max_{k} \{ y_k(\mathbf{x}) \} \right). \qquad (8)$$

If the second highest value c_{second} is equal to c_{max} the data is output as *not classified*. It is easy to change the algorithm to function with $c>2$ classes [20]. Usually three to seven epochs are needed for the whole training procedure.

The result of the training procedure are rules of the form (belonging to the core or support rectangle)

> **if** variable 1 **in** $(-\infty, 50)$ **and if** variable 2 **in** $(20,40)$
> **and if** variable 3 **in** $(-\infty,\infty)$ **then** *class l* (9)

in addition with a classification based on (8). Interestingly, in rule (9) variable 3 is not relevant, so variable 3 can be omitted and in such a case we get the simplified rule (10)

> **if** variable 1 **in** $(-\infty, 50)$ **and if** variable 2 **in** $(20,40)$ **then** *class l*
> (10)

How good are the resulting rules?

The relevance of a rule for a class can be measured by the number of samples of class k that lie in core (resp. support) rule p divided by the number of all samples. This is called the *frequency*. Additionally, the *class confidence* in a class decision is defined as the number of samples of class k that lie in p divided by the number of all samples that lie in p. Both measures, the class frequency and the class confidence of a rule, should always be calculated on test data samples, not on training data samples.

Using these two measures we can expand the rules to a more precise form. The expanded rule (10) becomes rule (11):

if variable 1 **in** (–∞, 50) **and if** variable 2 **in** (20,40) **then** *class* 1
     with *frequency* 5% **and** *class confidence* 80%   (11)

This concludes our tool set for extracting rule based knowledge of a data base.

4.2   Application to Septic Shock Patient Data

Now we present the results of the rule generation process of section 4.1 with the data set D of section 2. The data set D is 16-dimensional. A maximum of 6 variables for every sample was allowed to be missing. The missing values were replaced by random data from normal distributions similar to the original distributions of the variables. So it was assured that the algorithm can not learn a biased result due to biased replacements, e.g. means. We demand a minimum of 10 out of 17 variables measured for each sample, so there remained 1677 samples out of 2068 for analysis.

The data we used in 5 complete training sessions – each with a different randomly chosen training data set – was in mean coming from class 1 with a percentage of 72.10% and from class 2 with a percentage of 27.91%. In the mean 4.00 epochs were needed (with standard deviation 1.73, minimum 3 and maximum 7). Test data was taken from 35 randomly chosen patients for every training session, containing no data sample of the 35 patients in the training data set. In Table 5 the classification results are presented.

Table 5 Mean, standard deviation, minimum and maximum of correct classifications and not classifiable data samples of the test data set. In %.

	mean	standard deviation	minimum	maximum
correct classifications	68.42	8.79	52.92	74.74
not classified	0.10	0.22	0.00	0.48

Average specificity ("deceased classified / all deceased") was 87.96% and average sensitivity ("survived classified / all survived") was 18.15%. The classification result is not satisfying, although similar to the results in section 3.4 but with the benefit of explaining rules. Deceased patients

were not detected very well. Reasons for this can be the very individual behavior of the patients and the data quality (irregularity of measurements, missing values). In this way it seems not possible to classify *all* patients correctly, but it could be that in some areas of the data space the results are better (*local rules*). So we will present the results of the rule generation. In mean 22.80 rules were generated for class survived and 17.80 rules were generated for class deceased.

In Table 6 you can see the core and support frequencies resp. class confidences of the generated rules.

Table 6 Mean of frequency resp. class confidence of support and core rules (calculated on test data). In %. The average was taken from all repetitions and all rules of every repetition.

performance measure	class survived	class deceased
support frequency	15.93	13.33
core frequency	2.39	0.62
support class confidence	74.37	30.88
core class confidence	59.96	11.70

If no test data sample lies within a rule *p*, class confidence of *p* was set conservatively to zero, so that it is possible that the core class confidence could be lower than the support class confidence. All frequency values are in the normal range. Class confidence performance is not high, because there are a lot of *small* rules and a lot of rules containing samples of deceased *and* survived patients.

Despite these results it is possible to give some *single* rules with a better performance, e.g.:

if heart frequency **in** (105.00,∞) **and** systolic blood pressure **in** (130.00,∞) **and** inspiratorical O_2 pressure **in** (–∞, 60.00) **and** frequency of respiratory **in** (19.00,∞) **and** leukocytes **in** (–∞, 16.70) **and** dobutrex **in** (–∞, 1.50) **then** *class* survived **with** *frequency* 9.2% **and** class confidence 91.2% (containing data coming of 11 different patients)

if systolic blood pressure **in** (120.00,∞) **and** leukocytes **in** (24.10,∞) **and** dobutrex **in** (0.00, 6.00) **then** *class* deceased **with** *frequency* 7.6% **and** *class confidence* 69.7% (containing data of 13 different patients)

Considering the latter rule, we can present it to a medical expert in fuzzy notation after *defuzzification* (see [1]):

344

> **if** systolic blood pressure **is** high **and** (number of) leukocytes **is** high **and** dobutrex **is** given **then** patient **is** in a very critical condition

With the help of such rules, it may be possible for the physician to recommend therapies based on data analysis.

5   Conclusions and Discussion

The event of septic shock is so rare in the clinic routine that no human being has the ability to make a well-grounded statistical analysis just by plain experience. We have presented a data analysis approach for medical data and used it for the important problem of septic shock. The typical problems in analyzing medical data are presented and discussed. Although the special problem of septic shock diagnosis prediction is hard to solve the results of the basic analysis and the more advanced analysis by a growing neural gas are encouraging for the physicians to achieve an early warning system for septic shock patients, but our results are not final. In spite of severe restrictions of the data we achieved good results by using several preprocessing steps.

Our patient data of SIRS, sepsis and septic shock overlap heavily in the low-dimensional subspace we analyzed. Therefore, any prognostic system can not predict always the correct future state but may just give early warnings for the treating physician. These warnings constitute only an additional source of information; the backward conclusion that, if there is no warning there is also no problems, is not true and should be avoided.

Another diagnostic approach by neural networks is adaptive rule generation. By this, we can explain the class boundaries in the data and at the same time find out the necessary variables for the early warning system. By using a special approach of rectangular basis networks we achieved approximately the same classification results as by the growing neural gas. Additionally, the diagnosis was explained by a set of explicitly stated medical rules.

To see how difficult the problem of building an early warning system for septic shock patients is, we asked an experienced senior medical expert to propose an experience-based rule. The following rule was proposed:

if pH **in** (–∞,7.2) **and** arterial pO$_2$ **in** (–∞,60) **and** inspiratorical O$_2$ concentration **in** (80,∞) **and** base excess **in** (–∞,5) **then** *class* deceased

In fact, *no* data point of our data lies in the defined region: There is no data support for this opinion! So a rational data driven machine learning approach to metric rule generation is a great benefit in comparison with subjectively induced rules for the problem of septic shock.

Although the automatic rule generation approach is principally favorable, the number of 40 rules obtained is not much, but too much for daily clinical use. Here, much more research is necessary for selecting the most relevant rules and fusing a set of smaller, non-relevant rules to an efficient one. The performance measures class frequency and class confidence help, but do not solve these problems. In principal, we are faced with a principal problem: how do we get general rules if most of the samples are very individual ones, showing no common aspects? One solution to this fundamental problem is the search for new kinds of similarity. For instance, instead of static correlations or coincidences one might look for a certain dynamic behavior of the variables or their derivatives. In our case, small sampling frequencies and small data bases impeded such an approach.

The alternative to this weak diagnosis lies in the parallel analysis of all variables (in our case: about 140), not only a subspace of 16 in order to get rid of the overlappings and find good class boundaries in hyperspace. But here we encounter the important problem of "curse of dimensionality" [9] which is very hard to treat in the context of medical applications. Two main problems impede a successful approach: the small number of homogeneous patient data and the large number of missing values.

To improve our results we are collecting more data from septic shock patients from 166 clinics in Germany to evaluate our algorithms on this larger amount of patient data.

Generally, for both problems there is only hope if automatic data acquisition and exchange is available which is not the case in most hospitals in Europe. Nevertheless, by the introduction of cost controlling mechanisms (TISS-score etc.) hospital people are forced to enter all available data in the electronic patient record in order to get paid for their efforts. In turn, this may enable better analysis for us in near future by pushing

the change from the paper-and-pencil documentation style to electronic data acquisition systems.

There is another problem which should be mentioned here. Even if we have enough good quality data we encounter the problem of combining different kind of variables: metric variables like the one analyzed in this paper and categorical variables like operation and diagnostic code, drug prescription and so on. The transformation of each type into the other causes either an information loss or the introduction of additional, not justified information (noise). The standard approach to avoid this is the construction of an expert for each kind of data and to combine the output of both experts by a meta diagnosis, but there is no unifying approach for the analysis of both kind of data.

In near future we will try to improve the performance of these results by other methods. Further work will be a comparison of the achieved classification results with scores, which are known to have limitations in classifying individual patients (see [28]). Some results from cluster analysis are presented in [16].

Acknowledgements

References

[1]   Berthold, M. (1999), Fuzzy Logic, Chap. 8 in Berthold, M., Hand, D.J. (eds.) *Intelligent Data Analysis: An Introduction*, Springer-Verlag, 269-298

[2]   Berthold, M., Diamond, J. (1995), Boosting the Performance of RBF Networks with Dynamic Decay Adjustment. *Advances in Neural Information Processing Systems* 7, 521-528

[3]   Berthold, M., Diamond, J. (1998), Constructive Training of Probabilistic Neural Networks, *Neurocomputing* 19, 167-183

[4]   Brause R., Hanisch E. (2000), *Medical Data Analysis ISMDA 2000*. Springer Lecture Notes in Comp.Sc., LNCS 1933, Springer Verlag, Heidelberg

[5]   Brause, R. (1999), Revolutionieren Neuronale Netze unsere Vorhersagefähigkeiten? *Zentralblatt für Chirurgie* 124, 692-698

[6]   Brause, R., Friedrich, F. (2000), A Neuro-Fuzzy Approach as Medical Diagnostic Interface, *Proc. ESANN 2000,* 201-206, De Facto Publ., Brussels

[7]   Bruske, J. (1998), Dynamische Zellstrukturen. Theorie und Anwendung eines KNN-Modells. *Dissertation*, Technische Fakultät der Christian-Albrechts-Universität, Kiel, Germany

[8]   Bruske, J., Sommer, G. (1995), Dynamic Cell Structure Learns Perfectly Topology Preserving Map, *Neural Computation*, vol. 7, 845-865

[9]   Bellman, R. (1961) *Adaptive Control Processes: A Guided Tour*, Princeton, NJ: Princeton University Press

[10]   Dybowski, R. (1997), Assigning Confidence Intervals to Neural Network Predictions. *Technical Report, Division of Infection*,

UMDS (St Thomas' Hospital), London. 2 March 1997, available at http://www.umds.ac.uk/microbio/richard/nnci.pdf

[11] Fein, A.M. et al. (Eds.) (1997), *Sepsis and Multiorgan Failure*, Williams & Wilkins, Baltimore

[12] Fritzke, B. (1994), Fast Learning with Incremental RBF Networks. *Neural Processing Letters* 1(1) 2-5

[13] Fritzke, B. (1995), A Growing Neural Gas Network Learns Topologies, Proc. *Advances in Neural Information Processing Systems* (NIPS 7), in G. Tesauro, D. S. Touretzky, T. K. Leen, MIT Press, Cambridge, MA, 625-632

[14] Fritzke, B. (1997), Incremental Neuro-Fuzzy Systems, Proc. SPIE Vol. 3165, p. 86-97, *Applications of Soft Computing,* Bruno Bosacchi, James C. Bezdek, David B. Fogel (Eds.)

[15] Geisser, S. (1975), The Predictive Sample Reuse Method with Applications. *Journal of The American Statistical Association* 70, 320-328

[16] Hamker, F., Paetz, J., Thöne, S., Brause, R., Hanisch, E. (2000), Erkennung kritischer Zustände von Patienten mit der Diagnose "Septischer Schock" mit einem RBF-Netz. *Interner Bericht 04/00*, *Fachbereich Informatik*, J.W. Goethe-Universität Frankfurt a. M., http://www.cs.uni-frankfurt.de/fbreports/fbreport04-00.pdf

[17] Hartung, J. (1993), *Statistik: Lehr- und Handbuch der angewandten Statistik*. Oldenbourg-Verlag, München.

[18] Haykin, S. (1999), *Neural Networks, A Comprehensive Foundation*. Prentice Hall, 2nd edition, Upper Saddle River, NJ 07458

[19] Heinke, D., Hamker, F. (1998), Comparing Neural Networks, A Benchmark on Growing Neural Gas, Growing Cell Structures, and Fuzzy ARTMAP. *IEEE Transactions on Neural Networks* 9(6), 1279-1291

[20] Huber, K.-P., Berthold, M.R. (1995), Building Precise Classifiers with Automatic Rule Extraction. *IEEE International Conference on Neural Networks* 3, 1263-1268

[21] Inza I., Merino M., Larrañaga P., Quiroga J., Sierra B., Girala M. (2000), Feature Subset Selection Using Probabilistic Tree Structures. A Case Study in the Survival of Cirrhotic Patients Treated with TIPS, in [4], 97-110

[22] Kindermann, L., Lewandowski, A., Tagscherer, M., Protzel, P. (1999), Computing Confidence Measures and Marking Unreliable Predictions by Estimating Input Data Densities with MLPs. *Proceedings of the Sixth International Conference on Neural Information Processing* (ICONIP'99), Perth, Australia, 91-94

[23] Lavrač, N. (1999), Machine Learning for Data Mining in Medicine. In, Horn, W. et al (Eds.), *Proc. AIMDM`99. LNAI 1620*. Springer-Verlag Berlin Heidelberg, 47-62

[24] Martinetz, T. M., Schulten, K. J.(1994), Topology Representing Networks, *Neural Networks* 7, 507-522

[25] Members of the American College of Chest Physicians / Society of Critical Care Medicine Consensus Conference Committee (1992), Definitions for Sepsis and Organ Failure and Guidelines for the Use of Innovative Therapies In Sepsis, *Crit. Care Med.* 20, 864-874

[26] Mosteller, F., Tukey, J.W. (1968), *Data Analysis, Including Statistics*, in Handbook of Social Psychology 2, G. Lindzey und E. Aronson (Eds.), Addison-Wesley

[27] Nauck, D. (1999), Obtaining Interpretable Fuzzy Classification Rules from Medical Data. *Artificial Intelligence in Medicine* 16(2), 149-169

[28] Neugebauer, E., Lefering, R. (1996), Scoresysteme und Datenbanken in der Intensivmedizin - Notwendigkeit und Grenzen. *Intensivmedizin* 33, 445-447

[29] Paetz J., Hamker F., Thöne S. (2000), About the Analysis of Septic Shock Patient Data. In [4], 130-137 and at http://www.cs.uni-frankfurt.de/~paetz/PaetzISMDA2000.pdf

[30] Pietruschka, U., Brause, R. (1999), Using growing RBF nets in Rubber Industry Process Control, *Neural computing & Applications,* Springer Verlag, 8(2), 95-105

[31] Schumacher, M., Rößner, R., Vach, W. (1996), Neural Networks and Logistic Regression, Part I. *Computational Statistics & Data Analysis* 21, 661-682

[32] Seely A., Christou N. (2000), Multiple Organ Dysfunction Syndrome, Exploring the Paradigm of Complex Nonlinear Systems. *Crit. Care Med.* 28(7), 2193-2200

[33] Tagscherer, M., Kindermann, L., Lewandowski, A., Protzel, P. (1999), Overcome Neural Limitations for Real World Applications by Providing Confidence Values for Network Predictions. *Proceedings of the Sixth International Conference on Neural Information Processing (ICONIP'99),* Perth, Australia, 520-525

[34] Toweill D., Sonnenthal K., Kimberly B., Lai S., Goldstein B. (2000), Linear and Nonlinear Analysis of Hemodynamic Signals During Sepsis and Septic Shock, *Crit. Care Med.* 28(6), 2051-2057

[35] Vach W., Roner R., Schumacher M., (1996) Neural Networks and Logistic Regression: Part II, *Computational Statistics and Data Analysis* (21), 683-701

[36] Wade, S., Büssow, M, Hanisch, E. (1998), Epidemiology of Systemic Inflammatory Response Syndrome, Sepsis and Septic Shock in Surgical Intensive Care Patients, *Chirurg* 69, 648-655

[37] Wahba, G.; Wold, S. (1975), A Completely Automatic French Curve: Fitting Spline Functions by Cross-Validation, *Communications in Statistics* 4, 1-17

Appendix A: The network adaption and growing

Adaptation of the Layers

Let us input a multidimensional pattern $\mathbf{x}$ into the system. First, all neurons compare their match $\| \mathbf{w}_i - \mathbf{x} \|$ with that of the neighbors. That node b with the highest similarity, i.e. the smallest Euclidean distance between its weight vector and the input vector, will win the competition by its high activity y_i (*winner-takes-all*). There is also as second winner a node s with the second best match. Then, the weight vectors $\mathbf{w}_i$ in the neighborhood of the best matching node b are adapted by

$$\Delta \mathbf{w}_b = \eta_b \cdot (\mathbf{x} - \mathbf{w}_b)$$
$$\Delta \mathbf{w}_c = \eta_c \cdot (\mathbf{x} - \mathbf{w}_c) \qquad \forall c \in N_b \qquad \eta_b = 0.1, \ \eta_c = 0.01 \qquad (12)$$

as centers of Radial Basis Functions with the "step size" parameters η_b and η_c. In order to avoid rapid changes the new width $\sigma_i(k)$ of the bell-shaped functions are computed at time step k as shifted mean of the old values $\sigma_i(k-1)$ and the actual distances s_i

$$\sigma_i(k) = \gamma \cdot \sigma_i(k-1) + (1-\gamma) \cdot s_i \qquad \forall v_i \in G \qquad \gamma = 0.8 \qquad (13)$$

There is an error associated with each classification. This is defined as the Euclidean distance between the m-dimensional output vector $\mathbf{z}$ and the desired class vector $\mathbf{u}$ which has a one at dimension k if class k is desired as output and zero otherwise.

$$d(\mathbf{u},\mathbf{x}) = \| \mathbf{u} - \mathbf{z}(\mathbf{x}) \| \qquad (14)$$

The adaptation of the output weights is based on the delta rule [18] to decrease the error

$$\Delta w_{ji}^{out} = \eta_o (u_j - z_j) \, y_i \ ; \quad \forall \ j \in \{1,\dots,m\}, \quad \forall \ v_i \in G \quad \eta_0 = 0.01 \qquad (15)$$

Additionally, there is an error counter variable τ_i associated to every node v_i. The best matching neuron b stores the computed error of the output if the error is not marginal and exceeds a certain threshold θ_C.

$$\Delta \tau_b = \begin{cases} d(\mathbf{u},\mathbf{x}) & \text{if } d(\mathbf{u},\mathbf{x}) > \theta_C \\ 0 & \text{else} \end{cases} \qquad \theta_C = 0.2 \qquad (16)$$

All other error counters are exponentially decreased by

$$\Delta \tau_i = -\alpha \tau_i \qquad \forall \ v_i \in G \qquad \alpha = 0.995 \qquad (17)$$

Growing of the Representation Layer

In order to reduce the output error not only by adaptation but also by structural change, we insert a new neuron (new node) in the graph of the first layer. To do this, the node p with the highest error counter value is selected after a certain number (here:100) of adaptation steps. Between this node and its direct neighbor q with the highest error counter value a new node r is inserted. This new neuron receives a certain fraction β of the error of node p and the errors of p and q are decreased by β.

$$
\begin{aligned}
\tau_r &:= \beta \tau_p \\
\tau_p &:= (1-\beta) \tau_p \qquad\qquad\qquad \beta = 0.5 \qquad (18) \\
\tau_q &:= (1-\beta) \tau_q
\end{aligned}
$$

This cell growing allows us to start with a very small network and let it grow appropriately to the needs of the application. In comparison with other growing RBF nets (e.g. [30]) there is also a neighbor topology of edges. Each edge has an attribute called „age". According to this age, the edges may be deleted and update the topology of the graph.

- Increment the age of all edges [b , **.**] from the winner b by one.
- Reset the age of the edge between b and s to zero. If no edge between these nodes exists, create a new one with age zero.
- Delete all edges with age $\geq \theta_{age}$. $\theta_{age} = 60$
- Delete all nodes without an edge.

By insertion and center adaptation we control the construction of the network: regions with high error are increased while regions with no activity are decreased.

Appendix B: The main rule building algorithm

The parameters of the algorithm are (see Figure 6 and Figure 7):

$w_{k,i}$ weight of class k that is connected to neuron i,
$\mathbf{z}_{k,i}$ center of i-th rule prototype (= neuron) of class k,
$\lambda_{n,i,-}{}^{k}$ negative expansion of upper rectangle U,
$\lambda_{n,i,+}{}^{k}$ positive expansion of upper rectangle U,
$\Lambda_{n,i,-}{}^{k}$ negative expansion of lower rectangle L,
$\Lambda_{n,i,+}{}^{k}$ positive expansion of lower rectangle L

with *n* as data dimension, i=1, ...,m_1 with m_1= r for class k=1 and i=1, ..., m_2 with m_2= s for class k=2.

Reset weights:
for c = 1 **to** 2
   for i = 1 **to** m_c **do** $w_{c,i}$:= 0; $\lambda_{n,i,+-}$:= 0; $\Lambda_{n,i,+-}$:= ∞ **end**
end

Training of one epoch:
for each data sample **x** of class k **do**
   if $p_{k,i}$ covers **x**   // i.e. **x** lies in L
    then   // **x** is covered by $p_{k,i}$ (*cover*)
       $w_{k,i}$:= $w_{k,i}$ + 1;
       adjust $\lambda_{n,i,+-}{}^{k}$, so that U covers **x**;
       if x lies in a core rule U of a prototype of class c $\neq$ k
        then set all $\lambda_{n,i,+-}{}^{k}$:= 0; **end** // to prohibit over-
              lapping core-rules, additional to [20]
  Insert new neuron (commit):
    else
       m_k := m_k + 1;
       $w_{k,i}{}^{k}$:= 1.0;   // with i = m_k
       $\mathbf{z}_{k,i}{}^{k}$:= **x**;   // **x** is center of the new rule
       $\lambda_{n,i,+-}{}^{k}$:= 0;
       $\Lambda_{n,i,+-}{}^{k}$:= ∞;

  Shrink committed neuron:

       for c $\neq$ k, 1 $\leq$ j $\leq$ m_c **do**
        shrink $p_{k,i+1}{}^{k}$ by $\mathbf{z}_{c,j}$, i.e. shrink($p_{k,i+1}{}^{k}$, $\mathbf{z}_{c,j}$); // see app.C
       end
    end

Shrink conflict neurons:
for $c \neq k$, $1 \leq j \leq m_c$ **do**
    if x lies in support region L of p_j^c
        then shrink p_j^c by **x**, i.e. shrink(p_j^c, **x**);   // see app. C
    end
    end
end

Appendix C: The rule shrinking procedure

shrink(p,x) :
    p   one rule prototype,
    x   data sample,
    $z_{n,+-}$ center of the rule prototype (each dimension n is considered), left and right expansions are considered separately,
    $\sigma_{n,min}$ usually set to 0.1 (prohibits too small areas within one dimension)

- *minimal volume loss principle*:
calculate M for all *finite* Λ:
$M := \min\{ \ |z_{n,+-} - x_n| \ |$ for all $n \neq c$ and ...
    $... | \Lambda_{n,+-} - |z_{n,+-} - x_n| / \Lambda_{n,+-} | \ \leq \ | \Lambda_{c,+-} - |z_{c,+-} - x_c| / \Lambda_{c,+} \}$;
    if M exists **then** $\Lambda_{n,min,+-} := M$;
        if $M \geq \sigma_{n,min}$ **then** $\Lambda_{n,bestfinite,+-} := M$; **end**
    end

- calculate for all *infinite* expansions:
$N := \max\{ \ |z_n - x_n| \ | $ for all *n*$\}$;
    if N exists **then** $\Lambda_{n,max,+-} := N$;
        if $N \geq \sigma_{n,min}$ **then** $\Lambda_{n,bestinfinite,+-} := N$; **end**
    end

- Calculate a new $\Lambda_{n,+-}$ for p, i.e. a shrink in one dimension of the expansion:

      if $\Lambda_{n,bestfinite,+-}$ exists
        then $\Lambda_{n,+-} := \Lambda_{n,bestfinite,+-}$;
        else
          if $\Lambda_{n,bestinfinite,+-}$ exists **and** $((\Lambda_{n,bestinfinite,+-} \geq \Lambda_{n,min,+-})$ ...
                                              ... **or** $(\Lambda_{n,min,+-}$ does not exist))
            then $\Lambda_{n,+-} := \Lambda_{n,bestinfinite}$;
            else
              if $\Lambda_{n,min,+-}$ exists
                then $\Lambda_{n,+-} := \Lambda_{n,min,+-}$;
                else $\Lambda_{n,+-} := \Lambda_{n,max,+-}$;
              end
          end
      end

In the shrinking procedure, we added a threshold $\Lambda_{n,bestinfinite,+-}$ because $\Lambda_{n,min,+-}$ does not always exist. The original algorithm [20] can not be used with our real world data because the algorithm crashes, if not for all $n = 1, ...,m_c$ $\Lambda_{n,min,+-}$ exists, i.e. if for all n the relation $N < \sigma_{n,min}$ holds. If $\lambda > \Lambda$ for one of the λ's within a shrink procedure, set $\lambda := \Lambda$.