Combining statistical learning with a knowledge–based approach — A case study in intensive care monitoring

Katharina Morik Peter Brockhausen
Thorsten Joachims
Universität Dortmund, LS VIII
44221 Dortmund, Germany
{morik,brockhausen,joachims}@ls8.cs.uni-dortmund.de

April 27, 1999

Abstract

The paper describes a case study in combining different methods for acquiring medical knowledge. Given a huge amount of noisy, high dimensional numerical time series data describing patients in intensive care, the support vector machine is used to learn when and how to change the dose of which drug. Given medical knowledge about and expertise in clinical decision making, a first-order logic knowledge base about effects of therapeutical interventions has been built. As a preprocessing mechanism it uses another statistical method. The integration of numerical and knowledge-based procedures eases the task of validation in two ways. On one hand, the knowledge base is validated with respect to past patients' records. On the other hand, medical interventions that are recommended by learning results are justified by the knowledge base.

1 Introduction

In this paper, we want to present a challenging application of machine learning. The learning methods we use are already well known and theoretically well-founded. Why then should anybody be interested in our report on the application of these methods? Our principle point is that applications are a necessary precondition for the success of machine learning in several ways. On the one hand, practitioners who might want to apply machine learning benefit from scientific exploration of application fields. Well investigated application areas serve as points of reference so that experts of a field realize the impact of machine learning for their actual problems. The long way from learnability results to an application is made explicit step by step in the hope that these steps can easily be transferred to a similar practical problem. On the other hand, machine learning benefits from the challenge of true applications. Real world applications require many capabilities that can be ignored in learnability proofs or performance measurements with respect to a dataset library. First, detecting a suitable learning task in an application is one of the hardest problems when dealing with real-world applications. This task has been solved already for the datasets in, e.g. the UCI library. Second, another problem that is solved for the datasets in the libraries but demands much of the time when applying a learning algorithm to a new application is the feature selection or feature construction. Would there be more scientific interest in these topics if applications were of more concern [1]? Third, the use of background knowledge given by a domain expert determines our efforts in preparing the application. Can we easily write down what we have learned and is the learning algorithm able to make good use of it? Or do we have to find a clever way of how to encode the knowledge into our representation language? Fourth, the validation of machine learning results is an issue. When using a dataset of a library, accuracy of prediction is a suitable criterion. However, in real-world applications this criterion is one among others. Two other criteria are at least as decisive as is the accuracy, namely understandability and embeddedness.

By understandability, we denote how well an expert of the application domain can inspect the learning results in order to verify them. Most often, this demand restricts the representation formalism we can choose. Using a formalism which is close to the representation the expert is used to eases the verification by him or her. In addition, it restricts the size of the learning result. No expert has the time to inspect 10 or more pages of rules.

By embeddedness, we denote how well the learning algorithm can be integrated into the overall application system. This covers the use of already available data for learning as well as the use of learning results for processes of the application. The notions of pre- and post-processing are too much focused on the learning part of the application and, hence, simplify the issue of integrating several processes, among them possibly more than one learning algorithm. Again, the representation formalism for learning is constrained by the requirement of embeddedness. In contrast, the constraints regarding accuracy may even be weakened. In a sequence of (learning) processes, the low accuracy of one of them can be compensated by following (learning) processes[2].

Some of our arguments in favor of the scientific investigation of applications seem to underly the famous paper of Ross Quinlan (Quinlan, 1986). However, what received most of the attention was the ID3 algorithm. Also the subject of

[1]The application field of knowledge discovery has raised the interest in feature selection and construction again, cf. (Liu and Motoda, 1998).

[2]In our robotics application, we conducted a sequence of learning runs, where the first achieved only 71% accuracy. Although exactly these results were the input to the following learning run, the results of the last run showed 94% accuracy. Moreover, the robot could, in fact, navigate using the learned rules!(Klingspor, 1998)

multistrategy learning issued by Ryszard Michalski (Michalski and Wnek, 1997) seems to point into the same direction. However, there are also good points against application-oriented publications, which we do not want to ignore. First, it is hard to show that the chosen modelling of the application is optimal. If the customer is happy with it – wouldn't he be even happier with another approach? All we can do about this, is to make our raw data publicly available in order to allow for the reproduceability and comparability of results. Second, readers do not learn from yet another application of the same kind. Hence, we have to characterize our application such that principled new points become clear. This is what we try next.

We have investigated whether machine learning could enhance on-line monitoring in intensive care medicine. Why is this a challenging application? There are three groups of reasons, namely the data situation, the task of monitoring, and the particular constraints given by the application. Let us consider one after the other.

DATA SITUATION In modern intensive care, each minute several hundreds of measurements of a patient are recorded at the bed-side. This gives us a very high dimensional space of data about a patient. However, this does not mean that for each patient, each vital sign is recorded properly. The values of some vital signs are recorded only once within an hour. Some other vital signs are recorded only for a subset of the patients. Hence, the overall high dimensional data space is sparse. Moreover, the data is noisy with respect to the point in time whenever the protocol is made by nurses and not automatically. The average time difference between intervention as charted and calculated hemodynamic effect is 12.34 minutes for catecholamines, vasodilators, and rapid infusions – opposed to an expected time lag of very few minutes. Even the automatic measurements can be noisy, for instance, if somebody touches the tube or moves the bed. To make it even worse, some highly relevant parameters are not recorded at all, for instance, why the patient needs intensive care. To summarize, we have masses of noisy, high dimensional, sparse time series of numerical data.

Medical experts explain the numerical data in qualitative terms of high abstraction. The background knowledge given by the expert covers functional models of the human body as well as expertise in the proper treatment of intensive care patients including effects of drugs and volume input. In the experts' reasoning, time becomes the relation between time intervals, abstracting away the exact duration of, e.g., an increasing heart rate and focusing on tendencies of other parameters (e.g., cardiac output) within overlapping time intervals. To summarize, we have complex qualitative background knowledge explaining both the patient's and the doctor's behavior.

MONITORING The task of monitoring can best be understood as time-critical decision support. The final goal is to enhance the quality of clinical practice. This means that imitating the actual interventions, i.e. the doctor's behavior, is not the goal. Actual behavior is influenced by the overall hospital situation, e.g., how long is the doctor on duty, how many patients require attention at the same time. The goal of decision support is to supply the best recommendation under all circumstances (Morris, 1998).

CONSTRAINTS The application area of intensive care constrains our work in three ways. First, on-line monitoring restricts the computational efforts. The system that supports decision making must analyze many parameters and output a recommendation for an intervention – if necessary – in short real time. Second, experiments are not possible. We can only test our algorithms based on what we observe as the results of one particular intervention. Whether another intervention would have been better cannot be determined. Third, it must be easy to validate the acquired rules of decision both, with respect to previously unseen patient records and by experts' inspections. To summarize, the overall system must work in real time, taking numerical data as input and deliver recommendations in an understandable way.

Now, that we have explained why we consider applications a relevant subject for machine learning and our intensive care application a challenging one, we can start to describe our case study. The paper is organized as follows. First, we describe our layout for learning and validation: the detection of learning tasks, the modelling of background knowledge, the selection of learning algorithms. Second, we describe the learning tasks which the support vector machine has solved and how the results were evaluated. Third, we describe our modelling of medical knowledge in a restricted first-order logic and how we used reasoning and multistrategy learning for its validation. We then use the validated knowledge base for justifying interventions proposed by the learning results of the support vector machine. This additional validation of learning results is of particular importance when putting learning results to good use in medicine. We conclude by summarizing where we are and indicating our next steps.

2 Layout for Learning and Validation

Clinical decision support aims at providing doctors and nurses with therapy guidelines directly at the bed-side. This should enhance the quality of clinical care, since the guidelines sort out high value practices from those that have little or no value. The computerized protocol of care takes into account more aspects of the patient than a doctor can accommodate. It is not disturbed by circumstances or hospital constraints and it bridges the gap between low-level numerical measurements (the level of the equipment) and high-level qualitative principles (the level of medical reasoning). The system takes as input measurements of the patient as recorded at the bed-side. It outputs executable protocols of therapeutical interventions as a recommendation to the doctor. Such a decision support system has been developed and established at the LDS hospital of Salt Lake City for respiratory care (Morris, 1998). It is a knowledge-based system where the production rules have been acquired in about 25 person years. The system has been evaluated in several studies at diverse hospitals in more than 10 years. Our task is now to build such a decision

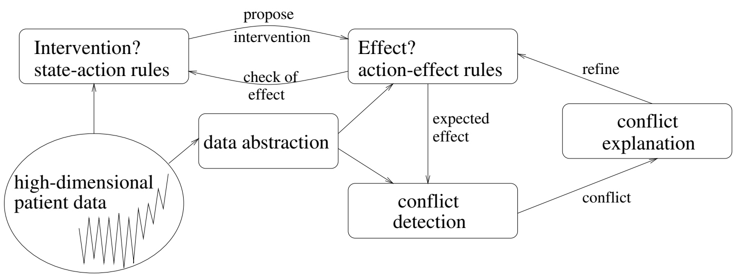

Figure 1: Overall architecture.

support system for hemodynamic care. The question is: can we achieve equally good results using much less resources (i.e. person years) if we apply machine learning?

Looking again at the list of advantages of computer-assisted intensive care, we obtain a list of requirements for the system to be built. The system must ground its decisions in explicit medical methods. We do not aim at modelling the hemodynamic system, the cardiac processes of patients. Neither do we aim at modelling the actual doctors' behaviors. Instead, the knowledge base must represent a therapy protocol which can be applied to measurements of the patient. This reminds us of the early days of knowledge acquisition for expert systems. However, our task at hand goes beyond classical medical knowledge acquisition, since the system has to cope with high dimensional data in real time. Its task is on-line monitoring, not heuristic classification or cover and differentiate. Moreover, the data consists of time series. Time stamped data do not necessarily require sequence analysis methods. For an application, we have to determine whether points in time, time intervals and their relations, or curves of measurements offer an adequate representation. How do we handle the patient's history? How do we summarize the curves of measurements to abstract qualitative propositions? These questions point at the problem of finding an adequate representation. Two sets of requirements on the capabilities of the representation can be distinguished:

1. The representation must handle numerical data, valid in one point in time, and time series. For each point in time, it must classify whether and which therapy intervention is appropriate for the patient.

2. The representation must handle relations of time intervals, interrelations of diverse drugs and relations between different parameters of the patient. It has to derive expected effects of medical interventions from medical knowledge and compare expected outcome with actual outcome.

The requirements are conflicting. Whereas we know good formalisms for each of the sets, we are not aware of a representation that fulfills both sets of demands. Hence, we decided to break down the overall reasoning into several processes and find an appropriate representation for each of them, independently. The overall architecture is shown in Figure 1. The patients' measurements at one point in time are used in order to recommend an intervention. This corresponds to clinical practice where for each point in time a recommendation for optimal treatment is needed. Of course, one of the recommendations is to not change the current therapy. The recommendation of interventions constitutes a model of *doctors' behavior*. A recommended intervention is checked by calculating its expected effects on the basis of medical knowledge. Medical knowledge qualitatively describes a patient's state during a time interval and effects of drugs. It constitutes a model of the *patients' hemodynamical system*. The medical knowledge base uses relations between time intervals and their abstract characterizations. To this end, patients' measurements are abstracted with respect to their course over time. The abstraction mechanism handles curves of measurements. The integration of numerical and knowledge-based methods allows us to validate the processes carefully. On one hand, the qualitative assessment of a statistical prediction enhances the model of the doctor's behavior in order to obtain a model of best practice. On the other hand, the medical knowledge is validated with respect to past patients' data. In detail, the processes we have designed are:

data abstraction Given series of measurements of one vital sign of the patient, eliminate outliers and find level changes. This abstracts the measurements to qualitative propositions with respect to a time interval, e.g., within time point 12 and time point 63, the heart rate remained about equal, from time point 63 to time point 69 it was increasing.

state-action rules Given the numerical data describing vital signs of the patient and his or her current medication, find the appropriate intervention. An intervention is formalized as increasing, decreasing or not changing the dose of a drug. The decision is made every minute.

action-effect rules Given the state of a patient described in qualitative terms, medical knowledge about effects of substances, relations between different vital signs, interrelation between different substances, a sequence of interventions, and a current intervention, find the effects of the current intervention on the patient. The derivation of effects is made for each intervention.

conflict detection Given the expected effect of a medication for a patient and his or her actual state, find inconsistencies.

conflict explanation Given interventions with effects on the patient that follow the medical knowledge and those that are in conflict with medical knowledge, find characterizations which separate the two sets.

It is straightforward to determine appropriate representations now: state-action rules and data abstraction use numerical functions, the other modules use a restricted first-order logic. Knowing the representation class for each process we can detect learning tasks. Which process can be modelled using machine learning? Which learning algorithm is appropriate for the task? For learning state-action

16	demographic attributes	11	vital parameters
5	intensive care diagnoses	37	I/O parameters
6	continuously given drugs	10	drugs
14	breathing parameters	10	laboratory tests
9	derived parameters		

Figure 2: Attributes in the hemodynamic dataset.

rules, we applied the support vector machine, because it is capable of handling high dimensional numerical data. Since the support vector machine is a binary classifier, we had to split the overall task of finding state-action rules into several particular learning tasks. For each drug, the support vector machine was trained on two tasks, namely learning the direction of interventions and learning when to intervene. This work is described in section 3. A statistical method for data abstraction was readily available (Imhoff et al., 1997). Action-effect rules were not to be learned, since our medical expert, Michael Imhoff, provided us with the medical knowledge. Using the MOBAL system (Morik et al., 1993) it was extremely easy to write the according rules. The inference engine of MOBAL derives expected effects and compares them with actual effects. These are deductive inferences. However, the explanation of conflicts between prediction and actual outcome requires to investigate many hypotheses. For this task, we used the inductive logic programming tool RDT/DB (Morik and Brockhausen, 1997). Work on action-effect rules and validation is described in section 4.

Data Set. The data set was collected at the 16-bed intensive care unit (ICU) of the "Chirurgische Kliniken der Städtischen Kliniken Dortmund". It contains the data of 147 patients between January 1997 and October 1998. Measurements are taken every minute, amounting to 679,817 observations for which data from a Swan-Ganz catheter is available. There are 118 attributes forming 9 groups (cf. figure 2).

Some of the parameters (like the demographic attributes) are not time dependent. While especially the vital parameters are measured on a minute to minute basis, others occur only once per hour or less.

3 Learning State-Action Rules

3.1 Support Vector Machine

Support vector machines (see (Vapnik, 1998)) are based on the *Structural Risk Minimization* principle (Vapnik, 1998) from statistical learning theory. The idea of structural risk minimization is to find a hypothesis h for which we can guarantee the lowest true error. Vapnik connects the bounds on the true error with the margin of separating hyperplanes. In their basic form support vector machines find the hyperplane that separates the training data with maximum margin. Since we will be dealing with very unbalanced numbers of positive and negative examples in the following, we introduce cost factors C_+ and C_- to be able to adjust the cost of false positives vs. false negatives. Finding this hyperplane can be translated into the following optimization problem:

$$\text{Minimize:} \quad \frac{1}{2}||\vec{w}||^2 + C_+ \sum_{i:y_i=1} \xi_i + C_- \sum_{j:y_j=-1} \xi_j \tag{1}$$

$$\text{subject to:} \quad \forall k: y_k[\vec{w} \cdot \vec{x_k} + b] \geq 1 - \xi_k \tag{2}$$

x_i is the feature vector of example i. y_i equals $+1$ (-1), if example i is in class $+$ ($-$).

We solve this optimization problem in its dual formulation using SVM^{light} [3] (Joachims, 1999a), extended to handle unsymmetric cost-factors. It can efficiently handle problems with many thousand support vectors, converges fast, and has minimal memory requirements.

3.2 Learning the Direction of Interventions

The first question we asked ourselves was: Given that we know the doctor changed the dosage of some drug, can we learn when he increased the dosage and when he decreased the dosage based on the state of the patient? To learn such a function, we had to first find an appropriate representation describing the patient's state.

3.2.1 What is an Appropriate Representation of the Patient's State?

Our dataset contains 118 attributes, some real valued, some categorial. Which features should we use for learning? How can we represent them appropriately for the SVM?

Categorial attributes are broken down into a number of binary attributes, each taking the values $\{0, 1\}$. Real valued parameters are either scaled so that all measurements lie in the interval $[0..1]$, or they are normalized by empirical mean and variance.

$$norm(X) = \frac{X - mean(X)}{\sqrt{var(X)}} \tag{3}$$

We systematically evaluated a large number of plausible feature sets using a train/test scheme. The feature set with the best performance is given in figure 3. According to a doctor, these are actually the most important parameters of the patient. Only the attributes "Cardiac Output" and "Net I/O" are missing, since they are seldomly present. Moreover, "Cardiac Output" measurements are always a risk for the patient.

[3] http://www-ai.cs.uni-dortmund.de/ svm_light

Vital Parameters	Continuously Given Drugs	Demographic Attributes
Diastolic Arterial Pressure	Dobutamin	Broca-Index
Systolic Arterial Pressure	Adrenalin	Age
Mean Arterial Pressure	Glyceroltrinitrat	Body Surface Area
Heart Rate	Noradrenalin	Ermergency Surgery?
Central Venous Pressure	Dopamin	
Diastolic Pulmonary Pressure	Nifedipin	
Systolic Pulmonary Pressure		
Mean Pulmonary Pressure		

Figure 3: The best feature set.

3.2.2 How much History is Necessary?

When making a decision about an intervention, it is (at least theoretically) possible to consider history in some form. We experimented with different ways of incorporating the history of the patient into the representation. We tried using only the measurements from one minute before the intervention (i. e. no history), using the last up to 10 minutes before the intervention, using averages of up to 60 minutes before the intervention, and combinations of both. We also tried incorporating the state of the patient at the previous intervention.

None of the approaches that use history gave significantly better results than just using the measurements from one minute before the intervention. This result is plausible and consistent with medical practice. According to doctors their decision to intervene is mostly based on the measurements they find after entering the patient's room. So short term history is ignored.

3.2.3 Prediction Performance

All experiments towards finding an appropriate representation were done on the training set. Using 10-fold cross validation on the training set we further optimized the parameters of the SVM (kernel and C). This lead to using linear SVMs for all drugs. The performance of the respective SVM on a previously untouched test set is given in figure 4.

To get an impression about how good these prediction accuracies are, we conducted an experiment with a doctor. On a subset of 40 test examples we asked an expert to do the same task as the SVM for Dobutamin, given the same information about the state of the patient. In a blind test the doctor predicted the same direction of dosage change as actually performed in 32 out of the 40 cases. On the same examples the SVM predicted the same direction of dosage change as actually performed in 34 cases, resulting in an essentially equivalent accuracy.

Drug	Accuracy	StdErr
Dobutamin	83.6%	2.5%
Adrenalin	81.3%	3.7%
Glyceroltrinitrat	85.5%	3.0%
Noradrenalin	86.0%	5.2%
Dopamin	84.0%	7.3%
Nifedipin	86.9%	7.0%

Figure 4: Accuracy in predicting the right direction of an intervention.

3.3 Learning when to Intervene

The previous experiment shows that SVMs can learn in how far drugs should be changed given the state the patient is in. In reality, the doctor also has to decide *when* to intervene or just keep a dosage constant. This leads to the following three class learning problem. Given the state of the patient, should the dosage of a drug be increased, decreased or kept constant? Knowing such a function is also a step towards deciding when to substitute one drug with another. Generating examples for this task from the data is difficult. The particular minute a dosage is changed depends to a large extent on external conditions (e.g. an emergency involving a different patient). So interventions can be delayed and the optimal minute an intervention *should* be performed is unknown. To make sure that we generate examples only when a doctor was closely monitoring the patient, we consider only those minutes where some drug was changed. This leads to 1319 training and 473 test examples.

For each drug we trained two binary SVMs. One is trained on the problem "increase dosage" vs. "lower or keep dosage equal", the other one is trained on the problem "lower dosage" vs. "increase or keep dosage equal". In order to better reflect the costs of inappropriate interventions, we use an SVM with a cost model. Lacking data for designing a more refined cost model, the cost-factors are chosen so that the potential total cost of the false positives equals the potential total cost of the false negatives. This means that the parameters C_+ and C_- of the SVM are chosen to obey the ratio

$$\frac{C_+}{C_-} = \frac{\text{number of negative training examples}}{\text{number of positive training examples}} \tag{4}$$

Figure 5 shows the test results for Dobutamin and Adrenalin. The confusion matrices give insight into the class distributions and the type of errors that occur. The diagonal contains the test cases, where the prediction of the SVM was the same as the actual intervention of the doctor. This accounts for 63% of the test cases for Dobutamin and for 79% of the test cases for Adrenalin. The SVM suggests the opposite intervention in about 1.5% for both drugs.

Again, we would like to put these numbers into relation with the performance of an expert when given the same information. For a subsample of 95 examples

Dobutamin	actual intervention		
	up	equal	down
predicted up	**46**	32	3
predicted equal	50	**197**	54
predicted down	5	30	**56**

Adrenalin	actual intervention		
	up	equal	down
predicted up	**23**	22	3
predicted equal	21	**310**	15
predicted down	4	34	**41**

Figure 5: Confusion matrices for predicting time and direction of Dobutamin and Adrenalin interventions.

Dobutamin	actual intervention		
	up	equal	down
predicted up	**10 (9)**	12 (8)	0 (1)
predicted equal	7 (9)	**35 (31)**	9 (9)
predicted down	2 (1)	7 (15)	**13 (12)**

Adrenalin	actual intervention		
	up	equal	down
predicted up	**4 (2)**	3 (1)	0 (0)
predicted equal	4 (6)	**65 (66)**	2 (2)
predicted down	1 (1)	8 (9)	**8 (8)**

Figure 6: Confusion matrices for predicting time and direction of Dobutamin and Adrenalin interventions in comparison to human performance.

from the test set, we again asked a doctor to perform the same task as the SVM. The results for Dobutamin and Adrenalin are given in figure 6. The performance of the SVM on this subsample is followed by the performance of the human expert (in brackets). Both are well aligned. Again, the learned functions of the SVM are comparable in terms of accuracy with a human expert. This also holds for the other drugs.

4 Action-Effect Rules

Based on detailed information from our medical expert, Michael Imhoff, we have built a compact knowledge base modelling the effects of drugs. Not counting patients' records, the knowledge base consists of 39 rules and 88 facts. Figure 7 shows a typical rule from the knowledge base. The rule states that increasing the dose

```
% Facts:
contains(dobutrex,dobutamin).
med_effect(dobutamin,1,10,hr,up).
med_effect(dobutamin,10,30,hr,up).
opposite(up,down).

% Rules:
intervention(P,T1,T2,M,D1) & intervention(P,T2,T3,M,D2) &
contains(M,S) &
med_effect(S,From1,To1,V,Dir) &
med_effect(S,From2,To2,V,Dir) &
ne(From1,From2) & gt(D2,D1) &
lt(D1,To1) & ge(D1,From1) & lt(D2,To2) & ge(D2,From2)
        → interv_effect(P,T2,T3,M,V,Dir).

% Patient Data:
level_change(pat460, 160, 168, hr, up)
intervention(pat460, 159, 190, dobutrex, 8)
```

Figure 7: Excerpt from the knowledge base.

from D1 to D2 of a drug M leads to an increasing effect on the parameter V of a patient P. The time intervals in which a certain dose is given to the patient are immediate successors. The dose is changed significantly.

4.1 Validating the Knowledge Base

In order to validate the knowledge base we applied it to the data of 148 patients. Part of an abstracted patient record is also shown in figure 7 for the time interval from the 160th minute to the 190th minute. Following an intervention, namely giving Dobutrex (= Dobutamin) in a dose of 8 units, a level change can be observed.

Overall, the patient data contain 8,200 interventions. 22,599 effects of the interventions were derived using forward chaining. In order to compare the predicted effects with the actual ones, we distinguish three types of conformity or contradiction. A predicted effect is

weakly conform with observed patient behavior, if no level change is observed, the patient's state remains stable;

strongly conform with observed patient behavior, if the observed level change has the same direction as is predicted by the rules;

contradictory with observed patient behavior, if a level change is observed into a direction opposite to the one predicted by the rules.

Note, that weak conformity is not in conflict with medical knowledge, but shows best therapeutical practice. Smooth medication keeps the patient's state stable and does not lead to oscillating reactions of the patient.

When matching the derived effects with the actual ones, the system detected:

weak conformity: 13,364 effects, i.e. 59.14%, took place in the restricted sense, that the patient's state remained stable.

strong conformity: 5,165 effects , i.e. 22.85%, took place in the sense, that increasing or decreasing effects of drugs on vital signs match corresponding level changes.

contradiction: 4,070 contradictions, i.e. 18.01% of the interventions, were detected. The observed level change of a vital sign went into the opposite direction of the knowledge-based prediction.

First, we started a knowledge revision process using concept formation using the methods of Stefan Wrobel (Wrobel, 1994). A concept is learned that separates successful rule applications (i.e. those, where the rules are not in conflict with the observations) from rule applications that lead to a contradiction. However, no clear separation could be found. Hence, we weakened the task to filtering out influential aspects. For learning, we chose 5,466 interventions with their effects being classified as conform (including the weak conformity described above) and as not conform. 11 predicates about the patient and the medications established the structured hypothesis space. Of all possible combinations, 121 hypotheses had to be tested. The findings were:

- The rule stating that a lowering a dose of a parameter increasing drug should lower the respective parameter is less reliable than the opposite rule.
- If combined with the age of the patient being around 55 years or the weight of the patient being small, the rule for effects of decreasing a medication is particularly unreliable.
- The weight of the patient alone has no impact on the reliability of action-effect rules.
- For elder patients (in the group of more than 65 years and in the group of more than 75 years), the weight is an influential feature.
- To our surprise, the amount of reducing or increasing the dose is not a relevant aspect for explaining contradictions, neither alone nor in combination with other features.

Relational learning did a good job in generating and testing many hypotheses. However, the learning results clearly indicate that the decisive features that would distinguish successful rule applications from not successful ones are not present in the data. The decisive features cannot even be formed from the available data

	art.	hr.	same effect all param.	same behavior
Noradrenalin	436	428	424	420
Dobutamin	403	395	383	299
Dopamin	472	472	472	387
Adrenalin	407	406	393	374
Glyceroltrinitrat	437	388	380	342
Nifedipin	457	457	455	438

Figure 8: Accuracy and equivalence of decisions.

using constructive induction! As is often the case, a negative result – even if it is well-based – is disappointing. It prevents us from inadequate revisions of the knowledge base, but it does not show us, how to effectively enhance the rules. When reporting our results to the medical expert, he assessed the ratio of 83.56 % correct predictions of effects very positively. Asked about possible missing data that could explain deviations, he indicated arhythmic heart beat as a decisive feature which is not present in the data. Also the missing values of cardiac output could possibly explain many deviations of observed from predicted effects.[4].

4.2 Using the Knowledge Base for Validating Interventions

As depicted in the overall architecture (cf. figure 1), we have chosen a design which allows us to use the action–effect rules in the knowledge base for validating predicted interventions. The underlying argument is that accuracy measures only reflect how well SVM's learning results fit actual behavior of the doctor. However, there are usually several different combinations of drugs that achieve the same goal of keeping the patient in a stable state. And indeed, different doctors, depending on their experience in the ICU, do use different mixtures and follow different strategies to reach this goal. For comparing treatment strategies, the real criterion is whether the recommendations have the same effect as the actual interventions. Therefore, we apply the action–effect rules from the knowledge base to both the proposed intervention of the SVM classifiers and to the intervention actually performed by the doctor. If the derived effects are equal, then the proposed decision of the SVM classifiers can be considered as "equivalent" to the intervention executed by the doctor.

The results of this comparison for 473 interventions are shown in figure 8. The right-most column indicates the accuracy, i.e. in how many cases the classification of SVM and doctor were identical (same behavior of SVM and doctor). The other columns state how often the SVM's intervention leads to the same effects as the intervention of the doctor. The first two columns show, how many of interventions

[4]Note, that cardiac output is not measured for all patients, because of its potential harm to the patient.

had the same effect on arterial blood pressure or heart rate, respectively. The third column gives a more concise evaluation. Here it is stated, how many interventions recommended by the SVM had the same effects on all vital signs as the actual intervention. For instance, the SVM correctly classifies 299 test cases for Dobutamin (63%). If we compare the resulting effects of the predicted interventions concerning Dobutamin with the effects of the actual doctor's interventions, we find that in 383 cases (81%) the deduced effects will be equal. Thus, in 84 cases the recommendation of the SVM does not match the doctor's behavior, but the effects are the same, since the doctor has chosen an "equivalent" drug or combination of drugs. This example demonstrates the advantage of our approach for validating learning results in contrast to merely looking at accuracy rates.

5 Conclusions

We present an application of machine learning for patient monitoring in intensive care. This application involves high dimensional time series data, demanding high quality decision support under real time constraints. It requires the integration of numerical data and qualitative knowledge. The tasks of reasoning are abstraction, classification, and deductive inference. These properties make this case study a representative for a large number of applications in medicine and engineering. Consider, for instance, robot applications, where measurements of the sensors and actions are to be integrated. Abstracting the measurements allows for high-level plans that cover a variety of situations. The classification of appropriate actions constitutes the low-level planning routines of the robot. If the costs of an inappropriate action are high, its justification on the basis of general knowledge is necessary. For instance, automatic car driving should integrate the low-level perception and action with general knowledge about the traffic law.

This paper presents the necessary steps for solving this application as a whole. We identify how the application can be split up into manageable parts. We propose an overall architecture that integrates a number of task, organized both sequentially and in parallel. All tasks are embedded in a single system, while selecting the most appropriate technique and representation – including the difficult task of selecting and constructing appropriate features – for each task individually. A statistical method is used to detect level changes in the curve of a patient's vital sign. The SVM is chosen for learning state-action rules due to its ability to handle many features. Several feature sets including the history of the patient were tested. Surprisingly, best results were achieved if only the patients' data one minute before an intervention were considered. This corresponds to the actual routine of a doctor. We present first experimental results demonstrating a performance comparable to that of a human expert in terms of accuracy. Moreover, the learned classifications of possible interventions are justified by deriving expected effects. This evaluation of the SVM's learning results goes beyond accuracy measurements and is much more realistic.

For modelling medical knowledge in terms of action-effect rules we chose a first order logic representation using MOBAL. This allows a compact representation of medical knowledge with a small number of rules, fulfilling the real-world demand for a knowledge base to be understandable to humans and accessible for expert validation. In addition, the knowledge base directly serves as background knowledge for learning refined rules and for doing knowledge revision. Discussions with experts in intensive care showed that the knowledge base is, in fact, understandable. We presented our results at the 9'th international symposium on intensive care (Joachims, 1999b; Morik and Imhoff, 1999) and received positive feedback on our modelling approach as being in-line with both the structure of medical knowledge and it's use in decision making. Moreover, the consistency checking of MOBAL allows to automatically detect cases where the actual patient state differs from the predicted effect of an intervention. Experts find it very useful to discuss a rule in the light of selected contradictory cases. The knowledge base turns the classifications of the SVM into operational knowledge for monitoring patients. The overall system is designed such that it can be applied at the hospital.The systemexploits patients' data as given and outputs operational recommendations for interventions. Hence, embeddedness guided the design process.

Our next step is the validation of the system by a committee of medical experts in order to further evaluate its performance. In particular, the combinations of different drugs need to be validated. A comparison with a hemodynamic knowledge base that is currently developed at the LDS hospital at Salt Lake City is planned. The LDS knowledge base does not take the stream of measurements as input, but reads vital signs on demand. It cannot be applied to past data and be evaluated with respect to them, because there is no component for checking consistency. We plan to transfer the knowledge base into our system so that it can be tested on patients' data. The impact of a stream of data (our approach) as opposed to some selected points in time when a vital sign is read (the LDS approach) will be investigated carefully.

Acknowledgements

This work has been funded in part by the collaborative research action on "Complexity Reduction in Multivariate Data Structures" (DFG, SFB475). Many thanks to Michael Imhoff, who leads the intensive care section of the city hospital of Dortmund, runs the clinical information system there, and nevertheless found time to explain us medical knowledge and practice. We also thank Ursula Gather and Markus Bauer for offering us their tool for time series analysis. We thank Jörg-Uwe Kietz for fruitful discussions about advantages and disadvantages of dataset libraries.

References

Imhoff, M., Bauer, M., Gather, U., and Löhlein, D. (1997). Time Series Analysis in Intensive Care Medicine. *Applied Cardiopulmonary Pathophysiology*, 6:203 – 281.

Joachims, T. (1999a). Making large-Scale SVM Learning Practical. In Schölkopf, B., Burges, C., and Smola, A., editors, *Advances in Kernel Methods - Support Vector Learning*. MIT-Press.

Joachims, T. (1999b). Wissenserlangung aus grossen Datenbanken. In W.Kuckelt and K.Hankeln, editors, *9th Int. Symposium on Indensive Care*, Journal f. Anaesthesie und Intensivbehandlung, Lengerich, Berlin, Riga, Scottsdale (Az.), Wien, Zagreb. Pabstscience Publishers.

Klingspor, V. (1998). *Reaktives Planen mit gelernten Begriffen*. PhD thesis, Univ. Dortmund.

Liu, H. and Motoda, H. (1998). *Feature Extraction, Construction, and Selection: A Data Mining Perspective*. Kluwer.

Michalski, R. and Wnek, J. (1997). Guest Editors' Introduction. *Machine Learning*, 27(3):205–208.

Morik, K. and Brockhausen, P. (1997). A Multistrategy Approach to Relational Knowledge Discovery in Databases. *Machine Learning Journal*, 27(3):287–312.

Morik, K. and Imhoff, M. (1999). Vergleich medizinischen Wissens und des aus einer intensivmedizinischen Datenbank gewonnenen Wissens am Beispiel des haemodynamischen Monitorings. In W.Kuckelt and K.Hankeln, editors, *9th Int. Symposium on Indensive Care*, Journal f. Anaesthesie und Intensivbehandlung, Lengerich, Berlin, Riga, Scottsdale (Az.), Wien, Zagreb. Pabstscience Publishers.

Morik, K., Wrobel, S., Kietz, J.-U., and Emde, W. (1993). *Knowledge Acquisition and Machine Learning - Theory, Methods, and Applications*. Academic Press, London.

Morris, A. (1998). Algorithm-Based Decision Making. In M.J.Tobin, editor, *Principles and Practice of Intensive Care Monitoring*, chapter 77, pages 1355 – 1381. McGraw-Hill.

Quinlan, R. (1986). Induction of Decision Trees. *Machine Learning*, 1(1):81–106.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley.

Wrobel, S. (1994). *Concept Formation and Knowledge Revision*. Kluwer Academic Publishers, Dordrecht.