Learning First Order Rules in Intensive Care Monitoring

Peter Brockhausen
Universität Dortmund, LSVIII, 44221 Dortmund, Germany
brockhausen@ls8.cs.uni-dortmund.de

Abstract

This paper describes a study on learning first order rules in noisy, real world, numerical time series data, describing patients in a intensive care unit. Given specific states a patient is in, the learned rules predict doctors' interventions to restabilize the patient. As a data preparation and abstraction method, statistical phase state models are used. They are used to transform the numerical signals, given on a minute by minute basis, into sequences of time intervals describing level changes. These new predicates are then used by the relational learner RDT/DB.

1 Introduction

Todays Clinical Information Systems (CIS) can provide the health care professional with the complete Electronic Patient Record (EPR) at the point of care. This data may include vital signs (e.g. heart rate, blood pressure), fluid intake and output, medications as well as plans of care, doctor's orders, and entire clinical pathways. These CIS are very complex database systems that comprise between several hundred and more than 2,000 variables for each patient (Imhoff, 1996; Imhoff, 1998).

On the one hand the doctor has access to all these variables at the bedside. But on the other hand we know that even experienced physicians are unable to develop solutions to a problem involving more than seven variables (Miller, 1956). What are the variables which influence the decision making at the bedside? If one asks the medical experts: "What is the central goal of therapy in an intensive care unit (ICU) concerning the hemodynamic system?" The answer will be: "Keep the patient in a stable state, or more precisely, the cardiac index should be greater than 4.5, mean arterial pressure between 90 to 100 etc."

The means to achieve this goal are combinations of six continuously given drugs and additional intake of infusions or output to control blood volume (the latter two will be ignored in the experiments). All these interventions have direct effects on the vital parameters, namely the heart rate, mean arterial blood pressure, mean pulmonary blood pressure, and central venous pressure.

The learning task we address in this paper is to discover state-action rules (Morik et al., 1999). Given the (critical) state of a patient (i.e at least one of the vital parameters is outside the interval defined by the respective norm values), what are the actions (i.e. interventions), which will take place to bring the patient back into a stable state.

In section 2 I give a detailed description of how to model the learning task to make it accessible for a relational learner. In section 3 I present and analyse the results before I draw a first conclusion and give an outlook on future work in section 4.

2 Data Preparation, Abstraction and Feature Construction

The data was collected at the 16-bed surgical intensive care unit of the "Chirurgische Kliniken der städtischen Kliniken Dortmund". The on-line monitoring data comprises the EPRs from

```
adrenaline(pat4606,1,up)
noradrenaline(pat4606,10,down)
adrenaline(pat4606,17,down)
not(noradrenaline(pat4606,1,up))
not(noradrenaline(pat4606,1,down))
not(adrenaline(pat4606,10,up))
not(adrenaline(pat4606,10,down))
not(noradrenaline(pat4606,17,up))
not(noradrenaline(pat4606,17,down))
```

Figure 1: Excerpt of goal predicates

148 consecutive critically ill patients (53 female, 95 male, average age 64.1 years) with extended hemodynamic monitoring requiring pulmonary artery catheters. These Swan-Ganz catheters allow measurements of the vital parameters used in the experiments on a minute by minute basis. In total, we have 680,332 observations[1].

Time series analysis was employed for data abstraction. Following previous studies (Imhoff et al., 1998), phase space models were used for this analysis. For the experiments we used models which were sensitive at a 5% level and with delayed-moving-windows of 15 minutes, i.e. the dynamic is modelled from the last 15 minutes (details can be found in (Bauer et al., 1999)).

Given series of measurements of one vital sign of the patient, this data abstraction delivers level changes. This transforms the quantitative signal to qualitative symbols, e.g. within time point 12 and time point 63, the heart rate remained about equal at one level. From 63 to 69 it has changed upwards to another stable state. As a side-effect, the phase space models also detect and eliminate outliers and artifacts in the data.

In this way we define time intervals for the vital signs heart rate, mean arterial pressure, mean pulmonary arterial pressure, and central venous pressure of each patient. For each time interval we compute the mean values and standard deviations. Medical knowledge tells us what the norm values of the four vital parameters should be. We construct new features, i.e. new predicates in our relational representation, which represent time intervals, e.g. where the mean of the heart rate was greater than the norm (e.g. `deviation(pat4606,5,13,hr,up)`). In the same way we build predicates where the mean values were greater (or lower) than the norm plus (or minus) the standard deviation.

Our goal predicates are the doctors interventions concerning the drugs Adrenaline, Noradrenaline, Dobutamine, Dopamine, Nifedipine, and Glycerol trinitrate. For each patient we collect all time points where the dose of at least one drug was changed. Then, we construct six sequences of positive and negative facts for all drugs. We have a positive fact, if a drug was changed and two negative facts, if it remains unchanged (figure 1 demonstrates the representation and transformation for two drugs).

To ease the learning by exploiting unification and to avoid unnecessary arithmetic comparisons, we replace all original deviation facts and split them in as many new deviation facts as interventions have taken place. In our example, we replace `deviation(pat4606,5,13,hr,up)` by the facts `deviation(pat4606,5, 10,hr,up)`, `deviation(pat4606,10,13,hr,up)`. As background knowledge, we have a predicate opposite with the obvious meaning.

For all experiments we use a 70 to 30 % split of the patients into a training and a test set. All parameter adjustments were only done on the training data. Table 1 summarises the amount of data we used for our experiments.

[1] In database terminology, we have one flat table with this number of records.

Table 1: Size of the data sets for learning and testing

	Training		Test	
	Pos	Neg	Pos	Neg
Adrenaline	777	29283	387	9255
Noradrenaline	546	29514	156	9486
Dobutamine	2427	27633	732	8910
Dopamine	261	29799	87	9555
Nifedipine	156	29904	78	9564
Glycerol trinitrate	1029	29031	420	9222
deviation	41536	—		

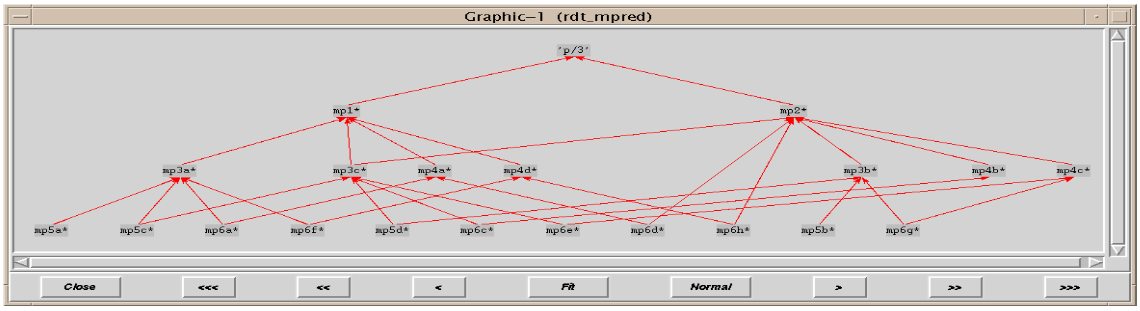

Figure 2: Metapredicate generalisation hierarchy.

3 Discovery of State-Action Rules

One of the central goals in intensive care medicine is to keep the patient in a stable state. But how can a stable state be characterised? The usual way is to define intervals of norm values, e.g. the heart rate should be between 60 to 100 beats per minute. Given the hypothesis, that the doctors at the ICU really try to reach this goal, we should be able to discover rules supporting this hypothesis. Our learning task is twofold and can be described as:

- how many and in which sequence have deviations to be present before an intervention will take place, and
- which kind of intervention, i.e. change of which drug, will take place.

This learning task is different from the one in (Morik et al., 1999), where we were interested in learning state-action rules too. In this paper, we are neither concerned with the direction of interventions nor the time points when to intervene. Instead, and in the spirit of discovery science, our interest is to find evidence in favour or against the hypothesis mentioned above.

For all learning experiments we used the learning system RDT/DB (Morik and Brockhausen, 1997), which is a variation of RDT (Kietz and Wrobel, 1992). RDT/DB uses for hypothesis generation the core of RDT, but for hypothesis testing, it translates hypotheses into SQL queries, which are executed and evaluated by an ORACLE database.

RDT/DB uses a declarative, syntactic bias for learning, called meta predicates. Figure 2 shows the metapredicate generalisation hierarchy, which was used for all experiments. The metapredicates cover situations with one to three deviations in sequence, in parallel or in combinations of both.

Figure 3 shows one example of a metapredicate and the corresponding rule that was learned. The metapredicate is a model for rules with two consecutive deviations (on different levels) of one

```
mp6d(P1, P2, P3, P4) : P2(P, T1, Te, Param1, Dir2) & P3(P, T2, T3, Param2, Dir1) &
                       P4(P, T3, Te, Param2, Dir1) & opposite(Dir1, Dir2) &
                       ne(Param1, Param2)
                       → P1(P, Te, Dir1).

deviation(P, T1, Te, Param1, Dir2) & deviation(P, T2, T3, Param2, Dir1) &
deviation(P, T3, Te, Param2, Dir1) & opposite(Dir1, Dir2) &
ne(Param1, Param2)
→ adrenaline(P, Te, Dir1).
```

Figure 3: Example of a metapredicate and a learned rule.

parameter and a third deviation of a different, second parameter. The former have to be in the opposite direction of the latter. The consequence is a change of the dosage of the drug Adrenaline into the same direction as the two consecutive deviations.

In medical domains (as in many others) it is nearly impossible to learn rules which are 100% correct. Therefore, one needs acceptance criteria for learning, which can deal with this situation. But usual criteria which only demand a high coverage or high accuracy are difficult to apply, because one does not know beforehand, where the limits will be. Thus, the demand that e.g. each rule has to cover at least 50% of the positive examples can be to high and one has to restart the learning process.

The acceptance and pruning criteria of RDT/DB consist of a user defined combination of elementary building blocks. They are combined with a likelihood ratio statistic to test significance of rules as in CN2 (Clark and Niblett, 1989). We are looking for rules which cover at least 5% of all positive facts, and where the ratio of covered positive facts to all covered facts is greater than the ratio of all positive facts to all facts. These rules have to be significant with an alpha value of 1%. In the notation of RDT/DB, where pos and neg are the numbers of covered facts of a rule, concl and negconcl the numbers of positive and negative instances of the conclusion predicates, this criterion is stated as:

$$\frac{pos}{pos + neg} > \frac{concl}{concl + negconcl} \quad \& \quad pos > 0.05 * concl$$

The likelihood ratio test will be computed as:

$$2 * \left(pos * log \left(\frac{pos}{pos + neg} * \frac{concl + negconcl}{concl} \right) + neg * log \left(\frac{neg}{pos + neg} * \frac{concl + negconcl}{negconcl} \right) \right)$$

In general, this requirement cannot be used for pruning, if it is not fulfilled. Using a top-down learner, specialization of a rule can make its likelihood ratio become significant, e.g. if they either cover only positive or negative facts. To prevent the learning of very special rules, the user can set the expected value for the minority class. If we have 100 positive and 100 negative facts and set the expected value to 5, then each rule must cover at least 10 facts, regardless of wether they are positive or negative. In combination with the required expected value, it is safe to use the outcome of the likelihood ratio test for pruning.

In total, we conducted four experiments. The first two use only the deviations of the mean values from the norm values. In the other two, we also took the standard deviation into account (cf. section 2). Both kinds of experiments were carried out with and without the likelihood ratio statistic. Instead of applying learned rules on the test set and simply evaluating the acceptance criterion, we start new learning runs and compare the resulting rule sets.

The first two experiments were unsuccessful. Using the likelihood ratio test, we could not learn any rules. Without the test, we learned five rules for three of the six drugs. But only one rule could

deviation(P,T1,T2,Param1,Dir1) → noradrenaline(P,T2,Dir1).
deviation(P,T1,T2,Param1,Dir2) & opposite(Dir1,Dir2) → noradrenaline(P,T2,Dir1).

deviation(P,T1,T2,Param1,Dir2) & opposite(Dir1,Dir2) → noradrenaline(P,T2,Dir1).
deviation(P,T1,Te,Param1,Dir1) & deviation(P,T2,Te,Param2,Dir1) & ne(Param1,Param2)
→ noradrenaline(P,Te,Dir1).

Figure 4: Rules learned for Noradrenaline.

be confirmed on the test data. One reason may be that we are too strict in considering deviations of the mean values from the norm values as our criterion for the construction of deviation facts. Although the intervals of norm values are rather large, there are patients with a permanently low or high blood pressure. Another reason may be that the norm values do not take into account the individual age, gender, and weight of a patient.

In the next two experiments, where we used the individual standard deviations of each time interval of each patient, our results were much better. For the drug Adrenaline, we learned exactly the same two rules on the training set in both experiments. In the experiment without the likelihood ratio test, we discovered the same two rules on the test set again. Using the test statistic, one rule learned on the test set was the same, the other one a direct specialisation.

In the case of Noradrenaline, using the likelihood ratio test results in some interesting differences. Without the test, we learned two conflicting rules on the training data. Only one of them was confirmed on the test data. In addition, we learned a second rule on the test data. This rule was more special than the rule, which was not confirmed, and it also resolves the conflict. Using the test statistic, we directly learned the two rules from the test set of the other experiment on the training set (figure 4 shows the four rules learned on the training data in this two experiments). In this case however, only one more special rule could be learned on the test set.

For the other drugs, the tendency was the same, using the likelihood ratio test during learning inhibits the acceptance of rules, which could not be confirmed on the test set. Our third and fourth experiment also showed that taking into account the standard deviations for predicate construction was the key to successful learning.

The time needed for each experiment was about 2 hours for six drugs. This includes both training and testing. The database run on a rather old Sun Sparcstation 20 with two CPUs a 50 MHz. Here, our RDT/DB approach really payed of. In comparison, RDT needed more than 20 times longer. Since we had more than 220,000 facts, advanced database techniques for joining large tables are much more efficient than simple first argument indexing, that PROLOG offers in the implementation of RDT.

4 Conclusion

In this paper we presented our approach to discovering state-action rules in medical data using ILP. One key to succesfull learning is the use of very advanced, statistical methods for feature, or here predicate, construction. Phase space models do not only deliver level changes needed for the symbolic description of numeric measurements. They also eliminate outliers and artifacts which are present in the data due to various reasons. In this way they also enhance the data quality. The other factor is the use of significance tests during learning, which are helpful to inhibit the learning of both, conflicting rules and rules which are only valid on the training data.

Using an ILP learning algorithm allows a very flexible and elegant handling of time intervals. The particular minute a drug is changed depends to a large extend on external conditions (e.g. an emergency involving a different patient). However, it is important, if there is a deviation of a vital parameter for some time interval from its norm values. Doctors react to these changes of the patient's state. Our representation language allows us to learn rules which express different temporal relations between deviations. In addition, (opposite) directions of deviations and directions of interventions are easily represented and exploited by unification of variables and background knowledge.

Another advantage of the relational representation is the understandability of the learned rules. Although we cannot elaborate on this point, the learning results presented here will be integrated into the overall system using MOBAL, presented in (Morik et al., 1999; Imhoff et al., 1999).

Although our learned rules make sense according to medical knowledge, a thorough validation by our medical experts is still needed. In this paper we analysed the question of which intervention will take place given some deviations of the vital signs. In some situations however, we know that it is mandatory to change more than one drug to bring a patient back into a stable state, e.g. due to arhythmic heart beat. Moreover, to prevent oscillating effects of vital parameters it may be necessary to change drugs more than once until the stable state is reached again. Handling these cases are some of the goals in our future works.

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 our colleagues Katharina Morik, Thosten Joachims, and Marcus Bauer.

References

Bauer, M., Gather, U., and Imhoff, M. (1999). Analysis of High Dimensional Data from Intensive Care Medicin. In Payne, R., editor, *Proceedings in Computational Statistics*. Springer. In print.

Clark, P. and Niblett, T. (1989). The CN2 Induction Algorithm. *Machine Learning*, 3(4):261–284.

Imhoff, M. (1996). 3 Years Clinical Use of the Siemens Emtec System 2000: Efforts and Benefits. *Clinical Intensive Care*, 7:43–44.

Imhoff, M. (1998). Clinical Data Acquisition: What and How? In W.Kuckelt and Hankeln, K., editors, *8th Int. Symposium Intensivmedizin, Intensivpflege*, volume 5 of *J. f. Anästhesie und Intensivbehandlung*.

Imhoff, M., Bauer, M., Gather, U., and Löhlein, D. (1998). Statistical Pattern Detection in Univariate Time Series of Intensive Care On-line Monitoring Data. *Intensive Care Medicin*, (24):1305–1314.

Imhoff, M., Gather, U., and Morik, K. (1999). Development of decision support algorithms for intensive care medicin: A new approach combining time series analysis and a knowledge base system with learning and revision capabilities. In *Proc. KI 99*. To appear.

Kietz, J.-U. and Wrobel, S. (1992). Controlling the Complexity of Learning in Logic through Syntactic and Task–Oriented Models. In Muggleton, S., editor, *Inductive Logic Programming.*, pp. 335–360. Academic Press.

Miller, G. (1956). The magical number seven, plus or minus two: Some limits to our capacity for processing information. *Psychological Review*, 63:81–97.

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

Morik, K., Brockhausen, P., and Joachims, T. (1999). Combining statistical learning with a knowledge-based approach – A case study in intensive care monitoring. In *Proc. 16th Int'l Conf. on Machine Learning (ICML-99)*, Bled, Slowenien. To appear.