Instance-Based Modelling in Medical Systems

Jiří Klema, Lenka Lhotská, Jiří Palouš, Olga Štěpánková
Department of Cybernetics, Faculty of Electrical Engineering,
Czech Technical University in Prague
Technicka 2, 166 27 Prague 6
email: {klema, step}@labe.felk.cvut.cz

Abstract

The paper tries to evaluate contribution of a special case of case-based reasoning, namely of instance-based modelling, to decision support in areas working with large amount of data. We concentrate on methodological questions of automated modelling and on advantages of our approach. It concerns namely description of the system life cycle, review of potential evaluation functions and division of data into training, testing and validating sets. Last but not least, the paper describes a real-life application of an instance-based modelling tool: we have complemented commercial CBR-Works 3.0 Professional system with an original module of automated interface. This module enables evolutionary optimisation of the model. The features of the resulting complex are shown on a case study concerned with prediction of result of Coronary Artery Bypass Graft (CABG) surgery operation.

1 Introduction

Case-based reasoning (CBR) [Kolodner, 1993] means adapting old solutions to meet new demands, using old cases to explain new situations, or to critique new solutions. CBR means reasoning from precedents to interpret a new situation or create an equitable solution to a new problem. CBR is able to utilize the specific knowledge of previously experienced, concrete problem situations (cases). A new problem is solved by finding a similar past case, and reusing it in the new problem situation. A second important feature is that CBR is an approach to incremental, sustained learning, since a new experience is retained each time a problem has been solved, making it immediately available for future problems.

If we watch the way people solve problems, we are likely to observe case-based reasoning in use all around us. Consider, for example, a doctor faced with a patient who has an unusual combination of symptoms. If the doctor has seen a patient with similar symptoms previously, he or she is likely to remember the old case and consider the old diagnosis as a solution to this new problem. If coming to that diagnosis was time-consuming in the earlier case, this method results in big savings of time. Of course, the doctor cannot assume the old answer is correct. He or she must still validate it for the new case in a way that does not prohibit considering other likely diagnoses. Nevertheless, quoting the old case allows the doctor to generate a plausible answer easily.

The CBR paradigm covers a range of different methods for organising, retrieving, utilising and indexing the knowledge retained in past cases. The term CBR is often used both as a generic one for several types of more specific approaches, and for one such approach. In its specific meaning, typical case usually has a certain degree of richness of information contained in it, and certain complexity with respect to its internal organisation. General background knowledge is used during reasoning process in order to modify, or adapt, a retrieved solution when applied in a different problem-solving context.

Nevertheless, the problem domain description in medical or other fields of study can often result in a standardised case description close to feature vector holding numeric values or symbolic values with simple inner structure. This situation can occur not only within shallow tasks. Even the huge and well-administered databases containing national registries correspond to a case structure that is not intricate enough to take advantage of all the upper mentioned CBR characteristics. On top of that, the solved tasks have very often quite simple definition space of the final solution. To persuade its future users about its usefulness, the system should first prove its predictive power when trying to answer unambiguous or one-dimensional questions before it is allowed to suggest more complex solutions. In particular, our ultimate goal is to build a system predicting the process of cardiological operation. But such a plan seems precocious before we can successfully predict whether to operate or not.

Then, is it still reasonable to use CBR in its generic meaning at least? Of course it is, as CBR can offer some comparative advantages. The first, CBR approaches are particularly suited for tasks where abstractions tend to yield over-generalisation. Additional benefit can be gained by using local information to characterise states and generate predictions. By retaining specific cases, decision can always be explained with aid of them. CBR represents lazy problem solving [Aha, 1998] where computation is performed on demand-driven basis. That is why, this approach is suited for incremental learning as well. Last but not least, CBR tolerates missing values, as it requires processing only the values known for the given query.

CBR theory [Aamodt and Plaza, 1994] denotes our view of automated modelling in medical systems as instance-based reasoning (IBR). IBR is a specialisation of another subclass of CBR - exemplar-based reasoning (EBR), which defines its concepts extensionally, as the set of all its exemplars. Solving a problem is a classification task. The set of classes constitutes the set of possible solutions. Modification of a solution found is therefore outside the scope of EBR. The IBR syntactic specialisation is based on simple representation of the instances. Moreover, IBR aims to study automated learning with no user in the loop. It is a non-generalisation approach to the concept learning problem addressed by classical, inductive machine learning methods.

2 Medical Systems

In medical decision support systems, general models of the given domain are usually generated. This approach works well, however in some cases it seems to be better if concrete past cases are available for decision making.

Comparing different machine learning (ML) approaches developed for answering the same question of prediction, neural networks offer usually the best solutions in terms of proposed evaluation function but they do not give any clue how the decision is formed. The decision trees are usually a little bit worse with respect to evaluation function calculated over all tested examples, but they come up with very lucid and understandable structure generating the final solution. The advantage of this tree is that it is general. On the other hand it can never be complex enough to answer all or at least reasonable part of expert's questions/doubts.

Instance-based learning seems to be a very natural approach to balance some of the above-mentioned disadvantages, as it is very similar to physician's notion of the given problem. The instance-based learning can answer both simple predictive questions and more complex questions of the whole process of treatment. Yes/no answers (or rather one-dimensional answers) applied e.g. in case of elective operations have to weight all risk factors that accompany the operation and post-operational recovery against risk factors connected with the failure caused by putting off the operation. The more complex type of answers seems to be even more suitable for instance-based reasoning as it gives the chance to take advantage of its knowledge-oriented procedures (and then it can be referenced as CBR). It can be utilised e.g. when physician searches for optimum process of operation and wants to follow past successful operations. Similarly, the instance-based reasoning can be used to avoid past questionable interventions.

2.1 Data Used in The Learning Process

ML theory suggests many different approaches how to deal with available data when model is generated. They can differ particularly in relation to the type of learning, which determines constant model characteristics, and size of data set itself. On all accounts, they follow two fundamental intentions: to enable generation of model with predictive power as high as possible and to give a chance to independently estimate its performance on future data and guarantee the model validity over these unseen data. In order to meet these two goals, all the data can not be used within model generation (adjusting model settings). The *hold-out method* divides data into two distinct sets: training set is used in learning and testing set is used in evaluation. *N-cross-validation method* divides the data into *N* partitions. Learning process runs in *N* steps, in each step *i* all the partitions except the i-th are used in learning, the i-th group is used for testing. *Leave-one-out method* is a special case of cross-validation, each partition consists of just one case and so number of learning steps is equal to number of data records. This method is very easy to be implemented with the instance-based techniques as it is trivial to ignore single case when searching for the most similar cases.

In recent years, attention is paid to generating and combining different but still homogenous classifiers with techniques called bagging, boosting or bootstrapping [Breiman, 1994, Dietterich and Kong, 1995]. They are based on repeated generation of the same type of model over evolving training data. These methods enable reduction of model variance. The authors prove that they can not be applied to instance-based learning cycle for the sake of method stability with respect to perturbations of the data. However, the idea of model variance reduction motivated our design of voting among several models based on the most promising settings.

2.2 Evaluation Function

Evaluation function is used to assess and differentiate the quality of solutions produced by different model settings and consequently it determines model refinement. Proper selection of the evaluation function seems to be key issue of the automated system design. Improper, superficial or schematic definition of the evaluation function brings undesired model bias and finally asks for time-consuming reiteration of experimental procedures. Generally speaking, the evaluation function should be objective, easily calculated and has to give a chance to determine uniquely an ordering of rated model settings. The common artificial intelligence and medical practice offers four different attitudes to overall quality evaluation of the solution produced by a model. Type of predicted variable and type of decision system output (that do not have to be necessarily same) determine their choice mainly.

The simplest attitude makes use of basic statistic functions calculated at a time over all training examples. *An average accuracy* defined as the relation between the correctly classified and all training objects can be used for this purpose for models that output categories. Similarly, the mean average absolute error can be used for model that are numeric. The advantage of this attitude lies in its simplicity; the disadvantage is that it can very often happen to be misleading.

The second approach coming from medical field separates the function to two distinct sub-measures - *sensitivity and specificity*. They are calculated in the same way as the average accuracy function (or other statistic functions), however sensitivity regards only positive patients (patients with observed final diagnosis) and on the contrary specificity deals only with negative patients (patients without observed final diagnosis). It is obvious that introduction of evaluation function division to these two distinct measures brings much more objectivity when dealing with unbalanced domain (the number of positive patients exceeds significantly number of negative patients or vice versa). On the other hand, the automated evolutionary development of the model usually asks for a single number to rank each model so that sensitivity and specificity should be used to design a unique measure anyway. However, if it is designed as a linear combination of both these sub-measures it can still keep the proportion between positive and negative patients. Obviously, the sensitivity and specificity can be applied if and only if when patients can be separated to two distinct mutually exclusive categories.

The next approach further refines the previous one. Each basic misclassification is assigned a level of importance – *differential misclassification cost*. The overall evaluation function is calculated as an average sum of weighted classification error costs. To be more specific, it can be laid down that it is much worse to predict an ill patient to be healthy than vice versa as this type of misclassification leads to the more extensive patient examination only. Opposite case can lead to patient examination delay and at the same time to serious consequences. The above mentioned explanation takes into the consideration only two possible decisions, as a matter of course the decisions can be much more complex and the error types can be valued by an error significance table. Although it is not absolutely obvious, very small changes in the error significance table can cause indispensable changes in model and intermediately in its behaviour. The special case of the error costs seems to be the classification to ordered set of classes (e.g. healthy, beginning illness, seriously ill) where the misclassification to adjoining class is not as serious error as assigning to distant class. The differential misclassification costs are employed by e.g. well-known algorithm C5.0 as well.

The fourth extensively used evaluation function we would like to mention is *a receiver operating curve (ROC) characteristic*. This measure comes to use mainly when it is not desirable to make the model predictions distinct, although the final classes are. The area under ROC gives a good chance to convert a quite complex and balanced comparison of all the predictions and real classifications to a single number. It can be shown that the area represents the probability that a randomly chosen diseased subject is correctly rated or ranked with greater suspicion than a randomly chosen non-diseased subject. In medical imaging studies the ROC curve is usually constructed as follows: images from diseased and non-diseased patients are thoroughly mixed, then presented in this random order to a decision system which is asked to rate each on a scale ranging from definitely normal to definitely abnormal [Hanley and McNeill, 1982]. The scale can be either continuous or discrete ordinal. The points required to produce the ROC curve are obtained by successively considering broader and broader categories of abnormal in terms of decision system scale and comparing the proportion of diseased and non-diseased patients.

In the section conclusion must be mentioned that above described evaluation process can become more complex with increasing complexity of the solutions produced by a model. More dimensional solutions consequently bring additional dimension to the evaluation process. These complex solutions furthermore ask for intelligent exploitation of the deep domain knowledge during evaluation function design.

3 CBR-Works 3.0

CBR-Works is a case-based reasoning system created in the frame of INRECA project. It is suited for intelligent solutions in a variety of domains and environments [CBR, 1998]. It includes the graphical editors that can support the user to design sophistically complex knowledge models. The system deals with concepts, types, similarity measures, weights and filters. Four separate interfaces provide a way to use elements for modelling concepts and types, case-base management, and case based retrieval:

- The concept hierarchy interface serves as an editor to build the concept part of the model,
- the types hierarchy interface serves as an editor to define the types and their similarity measures being used in the model,
- the CBR-Works case base interface provides the tools to manage the case base,
- the consultation interface offers operation to retrieve cases either by filling out a query or being guided by the query wizard.

The user can either define its own model elements or it can put in use modelling wizards and utilise standard predefined elements. The system is available on all major operating systems: Windows, UNIX, Macintosh (we have used Windows version).

However, when it comes to automated modelling or batch testing only, the on-line user interfaces become inefficient. Fortunately, CBR-Works uses a standardised exchange language especially developed for CBR applications called Case Query Language (CQL). It provides communication between CBR-Works servers and clients as well as it serves as interface language between the CBR-Works components. CQL is an object-oriented language for storing and exchanging the domain model description and cases in form of ASCII files. Furthermore, CQL is used for transportation of model and cases between CBR-Works servers and clients [CBR, 1998]. By means of CQL the system can be extended without restraint.

4 CBR Modelling Interface

CBR modelling interface [Palous, 2000] we have developed consists of four main conceptual units (see Figure 1). CQL wrapper is used to construct CQL queries from raw training/testing data and current experiment setting first, later it decomposes the CQL server responses to meaningful answers. Evaluation unit derives and accumulates the individual solutions, after all the training/testing examples are processed it calculates overall evaluation for the current evaluation setting. As a matter of fact it satisfies two distinct requirements: first it defines a way individual solutions are derived and for the second it identifies overall function in terms of previously defined theoretical demands. The experiment settings unit keeps current population of experiment settings and provides them gradually to the CQL wrapper during testing. The last evolutionary adaptation unit is responsible for generation of a new population of experiment settings at the end of each testing step.

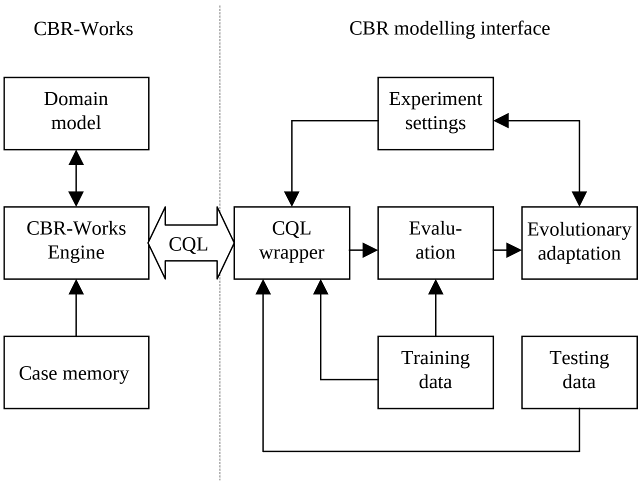

Figure 1 Automated modelling cycle

Construction of every domain model begins with initial definition of model settings within CBR-Works system by an expert (or with aid of background knowledge). He defines namely concept hierarchy and selects fitting concept similarity metrics. After that, case memory is loaded with feature vectors of selected patients. The patient vectors are transformed into CQL language first (stand-alone CQL_transformer is used for this purpose) and then imported into CBR-Works. Next, automated modelling can start. Different model settings are generated step by step and evaluated over training data using leave-one-out cross-validation. We employed *genetic algorithms* (GA) [Kubalik and Lazansky, 1998] and *sequential search* (SS) of restricted state space of possible parameter settings. Comparison of GA and SS depends mainly on level of restrictions of sequential search that determines how fast and exhaustive the second method is. Finally, the best model (the model with the highest value of evaluation function) is validated over testing (validating) data – in order to avoid overfitting, the model should show approximately the same value of evaluation function for the testing data as it shows for the training data set.

The previous paragraph regards but does not explain other important issues of IBR and automated modelling. There are many questions, e.g. which and how many patients are to be included into case memory, training and testing sets; can these sets change in iterative nature; shall we repeat the training cycle more times for different distribution of patient records among the data sets; what to do when the model proves to be overfitted and so on. Answers to all these questions are closely tied down to number of patients we deal with and time demands connected with each training cycle (experiments show that one query having tens of attributes to memory containing thousands of cases takes tens of seconds). The more patients we have the larger data sets we can create. The larger data sets are more time demanding when processed and that is why it is often hardly possible to iterate or repeat the training cycle. Moreover, they do not tend to produce overfitted models.

CBR interface has been developed in C++ Builder 3.0, MS Windows environment. The program is tabular and simply controllable, with minimum need of help. It communicates with CQL-server through Telnet protocol. The C++ library (namely TtnCnx component package) from *Internet Component Suite* package (freeware) is used for this purpose. It follows that the interface is able to connect to the server that can be anywhere in the Internet. Even thought the remote operation is not desirable when the model is tuned as it brings additional time delay it can be utilised for later consultation of its final version.

5 Case Study: Predictive Model Of Heart Operation Result

MEDICON Center is the center for development and operation of application and communication environment of the healthcare data network in the Czech Republic. It is intended to be a valuable resource for public, health outcome researchers, and academicians both in and outside of the Czech Republic. It is focused on establishment of the Merged National Registry (MNR) on Cardiovascular Interventions. Currently the MNR contains information on quality and results of diagnostic and therapeutic interventions done in selected relevant cardiovascular diseases. Currently the aggregated registry embodies 10.595 records. One record corresponds to one cardiovascular intervention. Each record consists of 160 attributes. The registry is run in environment of the informational system PATS (The Patient Analysis and Tracking System). PATS allows creation and administration of clinical user flexible databases. At the same time, it offers tools for long-term follow-up and statistical analysis pursuance. Well-known Bayesian technique is used for the purpose of short time patient oriented predictive statistics as well as for the purpose of general health care system predictive statistics that are done in long-term horizon [Medicon, 1998].

The Artificial Intelligence group at *Czech Technical University* was provided with the aggregate registry with an intention to use machine learning (ML) techniques for development of a tool which could enhance the PATS system predictive domain. The report [Klema *et al.*, 1998] presents preliminary experiments and their results. The research was aimed on Coronary Artery Bypass Graft (CABG) surgery, we have tried to construct a predictive model of heart operation result with aid of decision trees. The model predicted the result of patient operation (dead – living, nearmiss+ - nearmiss-) on the basis of the patient anamnesis and his/her pre-operative state. The methodology of decision tree construction proved to be rather inefficient for effectual distinction between successful and unsuccessful operations prior to its execution. We came to conclusion that the overall poor quality of decision-making was mainly caused by objective matter. All the information gains calculated during the process of the tree building showed that dependency of the final class on the other attributes is low. There was no single attribute exhibiting strong influence on the operation result. At the same time we draw conclusions that instance-based learning can bring additional gain when solving problems that are too hard to be described by highly generalised pieces of knowledge represented e.g. by a small set of rules or by a decision tree.

Three different classes were derived from two original attributes NEARMISS+ and STATUS. NEARMISS+ is the calculated parameter and it globally appreciates the quality of the intervention. It is a binary parameter, where 0 means intervention with a good result and 1 means intervention with a bad result. STATUS parameter denotes patient state after operation (1 – still alive, 2 – died during operation, 3 – died during hospitalisation, 4 – re-operation is necessary, 5 – died up to 30 days after operation, 6 – died more than 30 days after operation, 7 – death caused by a factor other than operation).

- Class 0 – NEARMISS+ is 0, arbitrary value of STATUS,
- Class 1 - NEARMISS+ is 1, STATUS differs from 2 and 3,
- Class 2 - NEARMISS+ is 1, STATUS is 2 or 3.

The other reason why the decision tree showed to be ineffective seems to be the unequal distribution of the patient set among the final classes. Figure 2 shows that more than 87% of the registry is assigned class 0, while only about 1% of examples belong to class 2. The classes 1 and 2 were classified with quite low accuracy that could be caused by noise and overfitting elimination technique. The classes rare in the training data can be more often affected by the pruning, as they are more inclinable to be misplaced with noise and consequently removed from the final tree structure. The evaluation function with differential misclassification costs brought improvement of the classification accuracy for the rare classes 1 and 2 but the proper setting of the costs which was made by hand showed to be problematic and highly time-consuming.

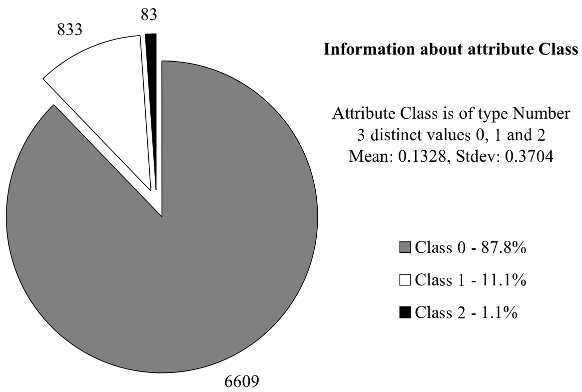

Figure 2 Class distribution within the MNR

The advantage of instance-based modelling interface lies in its flexibility. Block structure gives a good chance to change procedure of experiment with respect to problem domain characteristics. The most straightforward way of automated modelling is applicable just for the problem domains with many available records. The patient record set is randomly divided to two data sets – training and testing set. Distribution of patients among classes in both data sets should keep original distribution. The proportion of patients between data sets depends on time restrictions – the less time the automated modelling can take the higher number of patient records is inserted into the testing set. The training set is used simultaneously as case memory and training set in terms of Figure 1. The nearest case has to be always removed from query answer, as it is definitely the current training example. This technique we used to predict CABG operation result as well.

Evaluation unit derives the individual solutions in the form of class competence estimates. Each training example is assigned a probability vector defining its competence to classes 0,1 and 2. The vector values correspond to portions of individual classes among relevant cases and to the similarities of these cases to the given example.

The overall evaluation of the current model setting can be done in more ways. ROC characteristic is probably the most suitable one from the point of view of medical risk stratification. On the other hand, if we want to have a chance to compare IBR results to decision tree results the evaluation should be based on specificity and sensitivity. The maximum probability P_i is taken from every testing example vector and compared to its real classification. The overall quality of provided solution is derived of average match over the individual classes.

Experiments are very time consuming. So far we did not run any automated cycle with all the available data included in training or testing set and therefore we cannot compare our results to preliminary experiment outputs. The reduced tests were run both with GA and SS. The sequential search run on 3378 cases in memory, training set consisted of 55 examples. The automated setting tuning brought more than 25% increase of evaluation function, however reasonable part of this increase was caused by overfitting because of small training set. There were identified about 10 most relevant attributes (Age, BSA, Diabetes, PTCA, Pre-operation state, …) that roughly correspond to expert assumptions. Significance of less relevant attributes is never constantly low, consequently the attribute set proves to be hardly restricted. The final experiment is going to be run as soon as the reduced experiments suggest credible sectional procedures.

6 Conclusions

The paper deals with the field of instance-based reasoning. It addresses issues and assets connected with design and utilisation of systems employing such type of reasoning. We have introduced the CBR modelling interface that puts the hinted thoughts in practice. The CBR modelling interface extends CBR-Works system. It complements it by automated modelling features and creates full scope IBR system.

The features of the resulting system are shown on a case study concerned with prediction of result of Coronary Artery Bypass Graft (CABG) surgery operation. The prediction of CABG operation result prior the operation proves to be a hard task as the result is remarkably influenced by the course of the operation itself. Nevertheless, the suggested approach applies all its comparative advantages – adaptability with respect to its inner structure and consequently flexibility in its application and evaluation and ability to offer more information than other system types offer as relevant cases bring it anyhow. The experiments confirmed that possibility to optimise model parameters improves IBR performance so that it can be as good or even better than alternatives.

On the contrary, the indispensable disadvantage of the system lies in its enormous time requirements during the model tuning. The most time is spent by CBR Works system when searching for the most similar cases. We have already started to develop an alternative database subsystem that emulates all the necessary CBR Works features available through CQL. Preliminary experiments show that the speed-up of the non-optimised subsystem is two orders of the magnitude compared to the originally used CBR Works. The improvement seems to be promising as the subsystem can be further accelerated. The next suggested step is to design the patient database as distributed. Optimisation search techniques can be applied as well, although we are aware that sophisticated O(n log(n)) search techniques (e.g. k-d trees [Friedman *et al.*, 1977]) become quite inefficient in high dimensional spaces with dynamically changing search parameters.

Acknowledgments

The authors thank to MEDICON Center for providing us with Aggregate MNR data and consulting assistance. This research was supported by grant IG 3041/9/A from the Czech Technical University and grant CEZ: J04/98:210000012.

References

[Aamodt and Plaza, 1994] Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. *AI Communications*, IOS Press, Vol. 7: 1, pp. 39-59, 1994.

[Aha, 1998] Aha, D., W.: The Omnipresence of Case-Based Reasoning in Science and Application. A companion paper for an invited talk at *the 17th SGES International Conference on Knowledge Based Systems and Applied AI*, 1998.

[Breiman, 1994] Breiman, L.: Bagging predictors. Technical report 421, Department of Statistics, University of California, Berkeley, CA, 1994.

[CBR, 1998] CBR Works 3.0 documentation by TECINNO GmbH, Kaiserslautern, Germany, 1998.

[Dietterich and Kong, 1995] Dietterich, T.,G., Kong E.,B.: Machine Learning Bias, Statistical Bias, and Statistical Variance of Decision Tree Algorithm. Technical report available at http://www.cs.orst.edu/~tgd/ projects/, 1995.

[Friedman *et al.*, 1977] Friedman, J., H., Bentley, J.,L., Finkel, R., A.: An Algorithm for Finding Best Matches in Logarithmic Expected Time. ACM Trans. Math. Software, 3, pp. 209-226, 1977.

[Hanley and McNeill, 1982] Hanley, J., A., McNeill, B., J.: The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve. *Radiology 143*, p.29-36, April 1982.

[Klema *et al.*, 1998] Klema, J., Mikšovský, P., Štěpánková, O.: Predictive Model of Heart Operation Result Built on Merged National Registry on Cardiovascular Interventions of MEDICON Center. Research report GL 72/98, Prague, CTU FEL, 1998.

[Kolodner, 1993] Kolodner, J: Case-Based Reasoning, Morgan Kaufmann, 1993.

[Kubalik and Lazansky, 1998] Kubalík, J., Lažanský, J.: Genetic Algorithms and their Tuning. In Dubois, D. (Ed.), *Proceedings of CASYS '98*, Liege, Belgium, ISBN 1-56396-863-0, pp. 217-229, 1998.

[Lowe, 1995] Lowe, D., G.: Similarity Metric Learning for a Variable-Kernel Classifier, Neural Computation, 7, pp. 72-85, 1995.

[Medicon, 1998] MEDICON: Statistické nástroje NZR, IV. pracovní zasedání a WS MEDICON IKEM, 22.-23. 10. 1998.

[Palous, 2000] Palouš, J.: Využití případového usuzování pro lékařské aplikace. Diploma Thesis, FEL CTU, Prague, 2000.

[Wettschereck *et al.*, 1996] Wettschereck, D., Aha D., W., Mohri, T.: A Review and Empirical Evaluation of Feature Weighting Methods for a Class of Lazy Learning Algorithms, NCARAI Report: AIC-96-006, 1996.