DATA MINING STRATEGY SELECTION VIA EMPIRICAL AND CONSTRUCTIVE INDUCTION

Mykola Pechenizkiy
Department of Computer Science and Information Systems
University of Jyväskylä
P.O. Box 35
Jyväskylä 40351
Finland
mpechen@cs.jyu.fi

ABSTRACT
Nowadays there exist a number of data-mining techniques to extract knowledge from large databases. Recent research has shown that no single technique can dominate some other technique on all possible data-mining problems. Nevertheless, many empirical studies report that a technique or a group of techniques can perform significantly better than any other technique on a certain data-mining problem or a group of problems. Therefore, a data mining system has a challenge of selecting the most appropriate technique(s) for a problem at hand. In the real world it is infeasible to perform a comparison of all applicable approaches. Several meta-learning approaches have been applied for automatic technique selection by several researchers with little success. The goal of this paper is to consider and critically analyze such approaches. In the centre of our analysis we introduce a general framework for data mining strategy selection via empirical and constructive induction.

KEY WORDS
Data mining, meta-learning, constructive induction

1. Introduction

Data mining (DM) is an emerging area that considers the process of finding previously unknown and potentially interesting patterns and relations in large databases [1]. Numerous DM techniques have recently been developed to extract knowledge from these large databases. During the last years DM has evolved from less sophisticated first-generation techniques to today's cutting-edge ones. Currently there is a growing need for the next-generation knowledge discovery systems, which should be able to discover knowledge by combining several available techniques. It was pointed out in [2] that as classification systems become an integral part of an organizational decision support system (DSS), adaptability to variations in data characteristics and dynamics of business scenarios becomes increasingly important.
Michalski, with respect to the orientation of machine-learning systems, emphasized the importance of moving from single-strategy systems to development of systems that integrate two or more learning strategies [3]. The important issue here is that the integration should not be done in a predefined way. On the contrary, the goal is to develop a system that would integrate the whole spectrum of (machine) learning strategies and would be able to select the most suitable strategy or strategies for a given problem [3].
Selection of the most appropriate data-mining technique or a group of the most appropriate techniques is usually not straightforward. Since the DM process comprises several steps, which involve data selection, data pre-processing, and data transformation, applying of machine learning techniques, and interpretation and evaluation of patterns, selection of a technique for each step is needed. In this paper we refer to a combination of DM techniques selected as DM strategy.
In this paper we consider meta-learning approaches (Section 2) that use meta-data obtained from problem categorization and available techniques' categorization that have been applied for automatic technique selection by several researchers. Then we analyze the main limitations of such approaches, discuss why they were unsuccessful and suggest the way of their improvement, and introduce a general framework for DM strategy selection (Section 3). We conclude with the discussion of key issues and introduce the direction of further research (Section 4).

2. Automatic Algorithm Selection via Meta-learning

2.1 Meta-learning

Generally, it is hard to characterise algorithm's performance either as good or bad, or as better or worse than the performance of some other algorithm, since performance is context dependent. However, it is known from of many empirical studies that an algorithm or a group of some algorithms can perform significantly better than other algorithm on a certain problem or a group of problems that are characterised by some properties [2].
Meta-learning or bias learning is the effort to automatically induce correlations between tasks and inductive strategies. Thus, different DM approaches can be not only inductively evaluated within given domain but also correlated to domains' characteristics.

Successful meta-learning in the context of (automatic) algorithm(s) selection would be really very important and beneficial in the DM practice. It is obvious that in a real-world situation it is very unlikely to perform a brute-force search comparing all applicable approaches.
Several meta-learning approaches for automatic algorithm selection have been introduced in the literature. A comprehensive overview can be found in [4]. The most popular strategies for meta-learning are characterisation of a dataset in terms of its statistical/information properties or the more recently introduced landmarking approach [5] and characterization of algorithms.
The general idea of meta-learning with respect to selection of a classifier for a dataset at hand is presented in Figure 1.

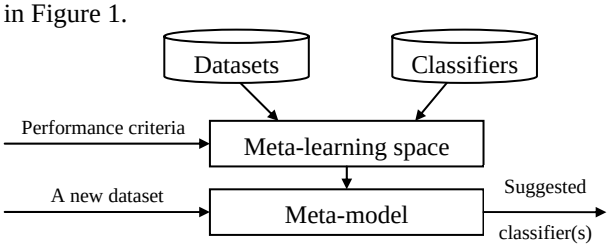

Fig. 1. Meta-learning system for suggestion of a classifier

Having a collection of datasets and a collection of classifiers, we can characterize them producing meta-data as a result of their (algorithms and datasets) characterisation. When meta-data is available, a machine-learning algorithm can be applied to it. As a result a meta-learning model that maps dataset characteristics to classifiers characteristics with respect to the introduced performance criteria is built. When a new dataset is introduced to the system, necessary dataset's characteristics are estimated so that the meta-model is able to suggest an algorithm or a combination of algorithms according to a performance criterion.
In the rest of this section we introduce the basic dimensions of automatic algorithm selection via meta-learning that include potential meta-learning goals, and meta-learning spaces, meta-data, and characterization of datasets and techniques available.

2.2 Meta-learning space

When providing a recommendation to a user, a suggestion may come as a list of applicable algorithms, the best algorithm or a ranking of the algorithms. In the first case the pool of classifiers is divided into two sets: algorithms that are expected to achieve good results on the dataset at hand, and algorithms that are expected to have poor performance [6]. In the second case the single algorithm that is expected to produce the best results on the dataset under examination, according to the performance criterion used, is presented in [7]. In the third case, the ranking list of potentially the most adequate algorithms can be suggested [8]. A good ranking algorithm should not only produce the ordered list of ranks but to check if the difference between algorithms is not significant and to show that fact if so (e.g. giving the same ranks).

In order to provide such recommendations, a corresponding meta-learning problem has to be established, since there is a need to learn something about the performance of learning algorithms. The meta-learning space (MLS) can be considered as a space that is constituted by collections of meta-learning problems [4]. Thus, when someone is interested in a list of applicable algorithms for a dataset, MLS consists of a number of (meta-)classification problems each of them is associated with one of the algorithms under consideration. Another formulation of MLS may include a number of meta-learning problems that correspond to all the pairwise comparisons of learners from a given pool. In this case meta-models describe the conditions under which one algorithm is preferable to another. A regression approach can be used also to predict the accuracy of an algorithm. Then, the MLS consists of one regression problem for each algorithm.

2.3 Characterisation of a dataset

Characteristics of a dataset that can be used for meta-learning are commonly divided into those that describe the nature of attributes, attributes themselves, associations between attributes, and associations between attributes and a target variable. To account for this, three categories of dataset characteristics can be used: (1) simple characteristics (number of examples, number of attributes, number of classes, etc.), (2) statistical characteristics (mean absolute correlation of attributes, mean skewness and mean kurtosis of attributes, etc.), (3) information theory characteristics (entropy of class, mean entropy of attributes, noise-signal ratio, etc.).
The first characterisation of a dataset in terms of its statistical/information properties can be referred to the framework of the Statlog project [9]. The exhaustive list of statistical and information measures of a dataset and the data characterisation tool (DCT) is proposed in [6].
The second approach to characterise a dataset is landmarking. The idea is to directly characterise a dataset by relating the performance of some learners (landmarkers) to the performance of some other algorithm [5]. This idea has reflection in the common practice of data miners. When a new problem is stated, some preliminary exploration is performed usually in order to quickly come up with some few potentially promising approaches.
Model-based data characterisation attempts to make use of properties of the concepts that are learned with certain algorithms or the direct use of the concepts themselves in a higher-order learning settings. In relational data characterisation different characteristics produced by the DCT from each attribute can be summarised by the histogram-based approach [4]. It was shown that histograms, when used as an aggregation method (e.g. mutual information between symbolic attributes), could preserve more information about the DCT properties of the individual attributes compared to the single aggregation functions (average, minimum and maximum).

2.4 Characterisation of an algorithm

Naturally, beside characterisation of a dataset from the data side, some application restrictions or priorities can be introduced. Thus, a user may like to define the most (un)desirable or the most crucial characteristic(s) of an algorithm to be selected for certain application. The most common characteristics that are taken into account are: algorithm taxonomy, interpretability of the model, importance of results interpretability and algorithm transparency, explanation of decision; training time, testing time, accuracy, and cost handling for misclassification.

A good overview of potential algorithm's characteristics is introduced in [10]. Three different types of sources, from which the knowledge of the characteristics of a learning algorithm is drawn, are considered. The first type includes basic requirements, capabilities and/or limitations of an algorithm. Corresponding characteristics usually are defined by the specification rather than determined empirically. Examples of such characteristics are algorithm run-time parameters, ability of handling misclassification costs, and data types supported. The second source is related to the expert knowledge about the algorithms. An interpretability example is a fact that a neural network results are assumed to be less interpretable comparing to associated rules; with respect to performance of algorithm an example could be the known fact that kNN is very sensitive to irrelevant attributes. The third source of algorithm characterization is the past learning experience. The potential problem with this source is that interactions of both the dataset and algorithm characteristics can distort the characteristics of the algorithm extracted from previous learning experience. From this perspective systematically controlled experiments might be more beneficial, since unlike real world datasets, synthetically generated data allow to test exactly the desired number of characteristics while keeping all the others unchangeable.

3. A Framework for DM Strategy Selection via Empirical and Constructive Induction

3.1 Limitations of meta-learning approaches

In the previous section we considered meta-learning approach as means of automatic algorithm selection, particularly, classifier selection for a dataset at hand. Despite of limited success reported by researchers, someone needs to admit that meta-learning approach as such has several shortcomings. Lindner and Studer in [6] reported on two general problems with meta-models that associate dataset and algorithm characteristics. The first problem is the representativeness of meta-data examples. The possible space of learning problems and thus a meta-learning-space is vast and getting larger with the invention of new algorithms, consideration of new characteristics and parameters. But the size of meta datasets used in the studies is naturally rather small because of computational complexity of producing a single meta-example – usually time-consuming cross-validation process is used to estimate the performance of every algorithm used in the study. The other problem (related especially to the landmarking approach) is computational complexity (up to $O(n^3)$, where n is the number of examples) of some sophisticated statistical measures. Such measures are not scalable to large datasets and actually such amount of resources can be used to a sophisticated learner directly.

In [11] a meta-learning framework that consists of coupling an IBL (kNN) algorithm with a ranking method was considered. The problem of algorithm selection/ranking has been divided into two distinct phases. During the first phase a subset of relevant datasets is identified by the introduced zooming technique. Zooming employs kNN with a distance function based on a set of dataset characteristics. In the second phase a ranking on the basis of the performance information of the candidate algorithms on the selected datasets that are similar to the one at hand is constructed. It was found that the selection of a ranking algorithm actually is much less important than the meta characteristics selection, especially when kNN (which is sensitive to irrelevant features) is used in the second phase. In [4] several meta-learners have been evaluated and compared. The C5.0 decision tree algorithm as a meta-learner performed better (non-significantly) than the other inductive algorithms. It seems however that so far there is no agreement among researcher on which strategy is most suitable for meta-learning. And although up-to-date published related papers show rather optimistic attitude to the meta-learning approach in general, the results presented in those papers are not so optimistic.

Most meta-learning approaches for automatic algorithm selection (meta-decision tree, meta-instance based learner) assume that the features used to represent meta-instances are sufficiently relevant. However, it was experimentally shown that this assumption often does not hold for many learning problems. Some features may not be directly relevant, and some features may be redundant or irrelevant. Even those meta-learning approaches that apply feature selection techniques, and can eliminate irrelevant features and thus some how account the problem of high dimensionality, often fail to find good representation of meta-data. This happens because of the fact that many features in their original representation are weakly or indirectly relevant to the problem. Existence of such features usually requires generation of new, more relevant features that are some functions of the original ones. Such functions may vary from very simple as a product or a sum of a subset of the original features to very complex as a feature that reflects whether some geometrical primitive is present or absent in an instance. The discretization (quantization) of continuous features may serve to abstraction of some features when reduction of the range of possible values is desirable.

The original representation space can be improved for (meta-) learning by removing less relevant features,

adding more relevant features and abstracting features. In the following section we consider a constructive induction approach with respect to classification in general and classifier selection in particular.

3.2 Constructive induction

Constructive induction (CI) is a learning process that consists of two intertwined phases, one of which is responsible for the construction of the "best" representation space and the second concerns with generating hypothesis in the found space [12].

In Figure 2 we can see two problems – with a) high quality and b) low quality representation spaces (RS). So in 'a)' the points marked by "+" are easily separated from the points marked by "–" using straight line or a rectangular border. But in 'b)' "+" and "–" are highly intermixed that indicates the inadequateness of the original RS. A traditional approach is to search for complex boundaries to separate the classes, and the constructive induction approach is to search for a better representation space where the groups are much better separated as in 'c)'.

Constructive induction systems consider learning as a dual search process for an appropriate representation in the space of representational spaces and for an appropriate hypothesis in the specific representational space. Michalski introduced constructive (expand the representation space by attribute generation methods) and destructive (contract the representational space by feature selection or feature abstraction) operators. Meta-rules construction from meta-data to guide the selection of the operators was considered.

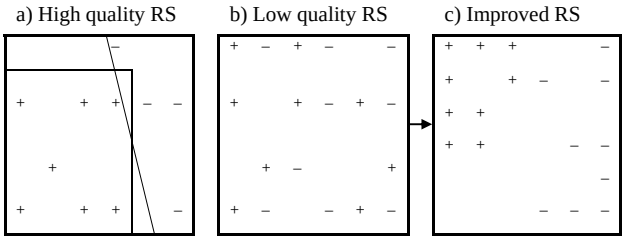

Fig. 2. High vs. low quality RS for concept learning [13]

Constructive induction methods are classified into three categories: data driven (information from the training examples is used), hypothesis driven (information from the analysis of the form of intermediate hypothesis is used) and knowledge driven (domain knowledge provided by experts is used) methods.

3.3 Framework for DM strategy selection

In this section we consider a general framework for DM strategy selection, which is depicted in Figure 3. This framework follows Fayyad's vision of KDD as a process and incorporates the idea of meta-learning for automatic algorithm selection, considered in Section 2 and constructive induction approach together.

KDD-Manager plays the central role here from the data-mining strategy selection point of view. Based on the meta-model that maps performance characteristics of available ML techniques and their combinations onto dataset characteristics KDD-Manager recommends the most appropriate algorithms that would constitute a certain DM strategy. Beside this main duty, KDD-Manager is responsible for the communication processes between different components (when cooperation and/or switching between different techniques is the case), and between an end-user or DM-researcher and the system.

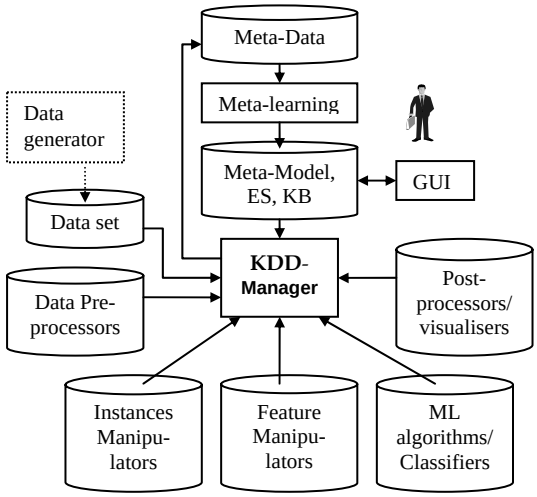

Fig. 3. DM strategy selection via meta-learning and taking benefit of constructive induction approach

Data Pre-processors are used for preliminary data cleaning, handling missing values, and that is more important from meta-learning perspective – retrieve the necessary characteristics of a *Dataset* at hand. A dataset can be an artificial one, synthetically produced by *Data Generator*. This possibility allows varying one group of dataset characteristics while controlling all the others (keeping them constant).

Machine-learning algorithms are used to construct classification or regression model or to cluster the representation space, producing thus local regions, so that further steps toward knowledge discovery can be applied locally for each region.

Instance manipulators and *Feature manipulators* are aimed to find the most suitable representation of the problem for a machine-learning algorithm. The possible scenario of data-mining strategy selection could be for example: k-means clustering, then ranked PCA-baised feature extraction (FE) [14], and then kNN classifier. An additional option is to apply the co-clustering technique that refers to simultaneous clustering along multiple dimensions, and particularly, in our case, along two dimensions: instances and features.

Post-processors and *visualizers* are aimed correspondingly, to enhance the induced models from both performance and interpretability perspectives (an example are rule grouping and pruning processes), and to present the achieved results to an end-user.

GUI (Graphical User Interface) is aimed to provide assistance for user input, recommendations, results reporting/visualisation/explanation, user feedback, and

Meta-model management.

Meta-model can be seen as the brains of KDD-Manager. A Meta-model is constituted by available expert knowledge, and knowledge acquired through meta-learning process on *Meta Data* that includes algorithms' and datasets' characteristics.

Data processing steps from an end-user point of view are similar to the ones introduced in [1]. The basic idea of a meta-learning cycle was considered in Section 2.1. However, we would like to be focused here on some important features of the framework that are implied by constructive induction.

Before applying a constructive induction approach, a decision about how relevant is the original representation space needs to be made. A landmarker approach can be used for this purposes or estimation may come directly from data characterization.

Generally, different evaluating criteria can be used for constructive induction with respect to the subsequent learning strategy. The gain ratio, gini index, and chi-square analysis are used when attributes are evaluated on the basis of their expected global performance, i.e. a new representation space is expected to discriminate among all classes. However, when each class is described independently from the other classes, construction of the simplest and the most accurate rules for each class is desired. Thus, a feature that has low global discriminating value (and thus ignored by a decision tree that uses a global criterion) at the same time may characterize very well a single class. Michalski in [12] shows this on example of recognizing the upper-case letters of English alphabet: feature *has tail* is very useful in the rule-based approach to discriminate *Q*s from the other letters, but this feature is most likely to be neglected since it has relatively low overall utility (because *Q*s normally occur rather rarely). When this is a case, a declarative knowledge representation may be more beneficial than the procedural one.

We consider FE for classification [14] as analogy of constructive induction. Indeed, what we are trying to achieve by means of FE is the most appropriate data representation for the subsequent classification. The notions of both destructive and constructive operators relevant to our approach and appropriate use of one or both of them can be beneficial. One approach is to select and perform FE keeping in mind the subsequent classification, estimate how good a new RS is and then perform selection of a classifier. This is the data driven approach. However, another approach – the selection of a combination of a FE technique and a classifier is more sophisticated. This is the hypothesis driven approach; in this case FE and classification cannot be separated into two different independent processes. Nevertheless, better understanding of the combined FE technique and a classifier as a whole and, respectively, better results on recommendation of a DM strategy that is constituted by the combination can be achieved.

It was shown that FE techniques are beneficial in cases with a very limited sample size [15] and highly correlated features [16]. Sohn in [8] shows that many meta-features that are commonly considered are highly correlated. Therefore we believe that dimensionality reduction of the meta-level by means of FE can provide better results. Furthermore, if case based reasoning (or just kNN) is used as a tool for meta-learning then the similarity measure is very important. Euclidian-based similarity depends heavily on how good is the instance space representation, namely, how relevant are the features throughout the whole space. Hence, better results can be achieved if (meta-) FE process is undertaken before a meta-learning technique is applied. Actually, we successfully applied a similar idea in some sense in [17], where FE is applied on meta-level for dynamic integration of classifiers.

4. Discussion and Concluding Remarks

In this paper we considered meta-learning approaches for automated algorithm selection and analyzed main limitations of such approaches, we discussed why they were unsuccessful and suggested the ways of their improvement.

In this section we summarise the key points of the proposed framework for DM strategy selection via empirical and constructive induction. Our main research goal is to contribute to knowledge in the problem of DM strategy selection for a certain data-mining problem at hand. Although there is a rather big number of DM techniques – one can hardly find a technique that would be the best for all datasets. This situation is very natural and it is typical with respect to any group of methods or techniques. For example, (if to consider all the datasets equally probable), it was proven in a number of so-called "no free lunch theorems" that for any algorithm, any improved performance over one class of problems is exactly paid for by performance over another class of problems [18]. Although this NFL theorem is often critisized on the fact that some of data sets are more probable to appear in the real world, even critics of this theorem would agree that for every method it's possible to find a data set (even if the probability to meet it as a real-life problem is very small) where its accuracy is less than that of a rival method(s). Thus, the adaptive selection of the most suitable DM techniques that constitute a DM strategy is a real challenging problem. Unfortunately, there does not exist canonical knowledge, a perfect mathematical model, or any relevant tool to select the best technique. Instead, a volume of accumulated empirical findings, some trends, and some dependencies have been discovered. On the other hand there are certain assumptions on performance of algorithms under certain conditions.

We proposed a DSS approach in the framework to recommend a data-mining strategy rather than a classifier or any other ML algorithm. And the important difference here is that constituting a data-mining strategy the system searches for the most appropriate ML algorithm with respect to the most suitable data representation (for this algorithm). In [17] we show particularly, how different

combinations of PCA-based FE techniques with a (meta-) classifier improve the performance due to the dual search. We believe that a deeper analysis of a limited set of DM techniques (particularly, FE techniques and classifiers) of the both theoretical and experimental levels is a more beneficial approach rather than application of the meta-learning approach only to the whole range of machine learning techniques at once. Combination of theoretical and (semiautomatic) experimental approaches requires the integration of knowledge produced by a human-expert and the meta-learning approach. In [19] we consider the approach based on a meta-learning paradigm as a way of knowledge acquisition from the experiments and on the methodology used in expert systems design for representation and accumulation and use of expert knowledge and knowledge acquired through the meta-learning process.

In the framework we considered the constructive induction approach that may include the feature extraction, the feature construction and the feature selection processes as means of relevant representation space construction.

We consider pairwise comparison of classifiers of the meta-level as more beneficial comparing to regression and ranking approaches with respect to contribution to knowledge, since the pairwise comparison gives more insight to the understanding of advantages and weaknesses of available algorithms, producing more specific characterizations.

With respect to meta-model construction we recommend the meta-rules extraction and learning by analogy rather than inducing a meta-decision tree. The argumentation is straightforward. A decision tree is a form of procedural knowledge. Since it has been constructed it is not easy to update it according to changing decision-making conditions. So, if a feature related to a high-level node in the tree is unmeasured (e.g. due to time-/cost-consuming processing), the decision tree can produce nothing but probabilistic reasoning. On the contrary, decision rules are a form of declarative knowledge. From a set of decision rules it is possible to construct many different, but logically equivalent, or nearly equivalent, decision trees [13]. Thus the decision rules are a more stable approach to meta-learning rather than the decision trees.

We would like to stress again on the possibility to conduct experiments on synthetically generated datasets that is beneficial from two perspectives. First, this allows generating, testing and validating hypothesis on DM strategy selection with respect to a dataset at hand under controlled settings when some data characteristics are varied while the others are held unchangeable. Beside this, experiments on synthetic datasets allow producing additional instances for the meta-dataset.

5. Acknowledgements

This research is partly supported by the COMAS Graduate School of the University of Jyväskylä, Finland. I would like to thank Dr. Alexey Tsymbal and Prof. Seppo Puuronen for their valuable comments and suggestions to this paper.

References:

1. U.M. Fayyad. Data Mining and Knowledge Discovery: Making Sense Out of Data, *IEEE Expert 11*(5), 1996, 20-25.
2. M. Kiang. A comparative assessment of classification methods, *Decision Support Systems 35*, 2003, 441-454.
3. R. Michalski. Toward a unified theory of learning: multistrategy task-adaptive learning. In *Readings in Knowledge Acquisition and Learning*, eds. B. G. Buchanan & D. C. Wilkins, 1993, 7-38.
4. A. Kalousis. Algorithm Selection via Meta-Learning. University of Geneve, PhD Thesis, Dept. of CS, 2002.
5. B. Pfahringer, H. Bensusan & C. Giraud-Carrier. Meta-Learning by Landmarking Various Learning Algorithms. *Proc. 17th Int. Conf. on Machine Learning*, San Francisco, 2000, 743-750.
6. G. Lindner & R. Studer. AST: Support for Algorithm Selection with a CBR Approach. *Proc. 3rd European Conf. on PKDD*, Prague, Springer Verlag, 1999, 418-423.
7. L. Todorovski & S. Dzeroski. Combining multiple models with meta decision trees, in: *Proc. 4rd European Conf. on PKDD*, Springer Verlag, 2000,. 54-64.
8. S. Sohn. Meta Analysis of Classification Algorithms for Pattern Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 21*, 1999, 1137-1144
9. D. Michie, D. Spiegelhalter, & C. Taylor. *Machine Learning, Neural and Statistical Classification*. (Ellis Horwood, 1994).
10. M. Hilario & A. Kalousis. Characterizing Learning Models and Algorithms for Classification. University of Geneva, TR UNIGE-AI-9-01, 1999.
11. C. Soares & P. Brazdil. Zoomed ranking: Selection of classification algorithms based on relevant performance information. *Proc. 4th PKDD*, Springer, 2000, 126-35.
12. R. Michalski. Seeking Knowledge in the Deluge of Facts, *Fundamenta Informaticae, 30*, 1997, 283-297.
13. T. Arciszewski, R. Michalski & J. Wnek. Constructive induction: the key to design creativity. TR MLI 95-6, George Mason University, 1995.
14. K. Fukunaga. *Introduction to statistical pattern recognition.* (Academic Press, London, 1999).
15. L. Jimenez & D. Landgrebe. High dimensional feature reduction via projection pursuit. PhD Thesis, TR-96-5, 1996.
16. I. Jolliffe. *Principal Component Analysis*. (New York: Springer-Verlag. 1986)
17. A. Tsymbal, M. Pechenizkiy, S. Puuronen & D. Patterson 2003. Dynamic integration of classifiers in the space of principal components. *Proc. 7th East-European Conf. ADBIS'03*, Heidelberg: Springer-Verlag, 2003, 278-292.
18. D. Wolpert & W. MacReady. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation, 1*(1), 1996, 67-82.
19. M. Pechenizkiy, S. Puuronen & A. Tsymbal. Feature Extraction for Classification in the Data Mining Process. *Information Theories and Applications 10*(1), Sofia, FOI-Commerce, 2003, 321-329.