FEATURE EXTRACTION FOR CLASSIFICATION IN THE DATA MINING PROCESS

M. Pechenizkiy, S. Puuronen, A. Tsymbal

Abstract: Dimensionality reduction is a very important step in the data mining process. In this paper, we consider feature extraction for classification tasks as a technique to overcome problems occurring because of "the curse of dimensionality". Three different eigenvector-based feature extraction approaches are discussed and three different kinds of applications with respect to classification tasks are considered. The summary of obtained results concerning the accuracy of classification schemes is presented with the conclusion about the search for the most appropriate feature extraction method. The problem how to discover knowledge needed to integrate the feature extraction and classification processes is stated. A decision support system to aid in the integration of the feature extraction and classification processes is proposed. The goals and requirements set for the decision support system and its basic structure are defined. The means of knowledge acquisition needed to build up the proposed system are considered.

Keywords: Feature Extraction, Classification, Data Mining.

Introduction

Data mining applies data analysis and discovery algorithms to perform automatic extraction of information from vast amounts of data. This process bridges many technical areas, including databases, human-computer interaction, statistical analysis, and machine learning.

A typical data-mining task is to predict an unknown value of some attribute of a new instance when the values of the other attributes of the new instance are known and a collection of instances with known values of all the attributes is given. In many applications, data, which is the subject of analysis and processing in data mining, is multidimensional, and presented by a number of features. The so-called "curse of dimensionality" pertinent to many learning algorithms, denotes the drastic raise of computational complexity and classification error with data having high amount of dimensions [Bellman, 1961]. Hence, the dimensionality of the feature space is often reduced before classification is undertaken.

Feature extraction (FE) is one of the dimensionality reduction techniques [Liu, 1998]. FE extracts a subset of new features from the original feature set by means of some functional mapping keeping as much information in the data as possible [Fukunaga, 1990]. Conventional Principal Component Analysis (PCA) is one of the most commonly used feature extraction techniques. PCA extracts the axes on which the data shows the highest variability [Jolliffe, 1986]. There exist many variations of the PCA that use local and/or non-linear processing to improve dimensionality reduction [Oza, 1999], though they generally do not use class information.

In our research, beside the PCA, we discuss also two eigenvector-based approaches that use the within- and between-class covariance matrices and thus do take into account the class information. We analyse them with respect to the general task of classification, to the learning algorithm being used and to dynamic integration of classifiers (DIC).

During the last years data mining has evolved from less sophisticated first-generation techniques to today's cutting-edge ones. Currently there is a growing need for next-generation data mining systems to manage knowledge discovery applications [Fayyad, 1996a]. These systems should be able to discover knowledge by combining several available data exploration techniques, and provide a fully automatic environment, or an application envelope, surrounding this highly sophisticated data mining engine [Fayyad, 1996b].

In this paper we consider a decision support system (DSS) approach [Turban, 2001] that is based on the methodology used in expert systems (ES) [Jackson, 1999]. The approach combines feature extraction techniques with different classification schemes. The main goal of such a system is to automate as far as possible the selection of the most suitable feature extraction approach for a certain classification task on a given data set according to a set of defined criteria.

In the next sections we consider the feature extraction process for classification and present the summary of achieved results. Then we consider a decision support system that integrates the feature extraction and classification processes, describing its goals, requirements, structure and the ways of knowledge acquisition. As a summary the obtained results are discussed and the focus of the further research is described.

PCA-based Feature Extraction

Generally, feature extraction for classification can be seen as a search among all possible transformations of the feature set for the best one, which preserves class separability as much as possible in the space with the lowest possible dimensionality [Fukunaga, 1990]. In other words we are interested in finding a projection w:

$$\mathbf{y} = \mathbf{w}^T \mathbf{x} \qquad (1)$$

where y is a $p' \times 1$ transformed data point (presented using p' features), w is a $p \times p'$ transformation matrix, and x is a $p \times 1$ original data point (presented using p features).

In [Oza, 1999] it was shown that the conventional PCA transforms the original set of features into a smaller subset of linear combinations that account for the most of the variance of the original data set. Although it is the most popular feature extraction technique, it has a serious drawback, namely the conventional PCA gives high weights to features with higher variabilities irrespective of whether they are useful for classification or not. This may give rise to the situation where the chosen principal component corresponds to the attribute with the highest variability but having no discriminating power.

A usual approach to overcome the above problem is to use some class separability criterion [Aivazyan, 1989], e.g. the criteria defined in Fisher linear discriminant analysis and based on the family of functions of scatter matrices:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \qquad (2)$$

where S_B is the between-class covariance matrix that shows the scatter of the expected vectors around the mixture mean, and S_W is the within-class covariance, that shows the scatter of samples around their respective class expected vectors.

A number of other criteria were proposed in [Fukunaga, 1990]. Both parametric and nonparametric approaches optimize the criterion (2) by using the *simultaneous diagonalization algorithm* [Fukunaga, 1990].

In [Tsymbal, 2002] we analyzed the task of eigenvector-based feature extraction for classification in general; a 3NN classifier was used as an example. The experiments were conducted on 21 data sets from the UCI machine learning repository [Blake, 1998]. The experimental results supported our expectations. Classification without feature extraction produced clearly the worst results. This shows the so-called “curse of dimensionality” with the experimented data sets and the necessity to apply feature extraction with them. The conventional PCA was the worst feature extraction technique on average. The nonparametric technique was only slightly better than the parametric one on average. However, this can be explained by the selection of the data sets, which are relatively easy to learn and do not include significant nonnormal class distributions. Besides, better parameter tuning can be used to achieve better results with the nonparametric technique. The nonparametric technique performed much better on categorical data for this selection of the data sets.

Still, it is necessary to note that each feature extraction technique was significantly worse than all the other techniques at least on a single data set. Thus it was shown that among the tested ones there does not exist any “the overall best” feature extraction method for classification with regard to all given data sets, and the problem of selection of the best suited feature extraction algorithm with its optimal parameters for classification was stated.

Feature Extraction for a Classifier and Dynamic Integration of Classifiers

The other interesting research question is to look for the best combination of a feature extraction method and a classifier among the available methods for a data set. We considered three PCA-based feature extraction methods with a number of different classifiers. A series of experiments were conducted on the same 21 data sets from the UCI machine learning repository. The results showed that there does not exist “feature extractor – classifier” pair that would be the best one for any given data set.

The other problem of search for the best suited feature extraction algorithm and its parameters for a certain classifier with regard to the given data set was stated.

Recent research has proved the benefits of the use of ensembles of classifiers for classification problems [Merz, 1996]. The challenge of integration is to decide which classifier to select or how to combine classifications produced by several classifiers.

The integration of an ensemble of classifiers has been shown to yield higher accuracy than the most accurate base classifier alone in different real-world tasks. The two main approaches to the integration are: first, the *combination approach,* where the base classifiers produce their classifications and the final result is composed using those classifications and second, the *selection approach,* where one of the classifiers is selected and the final result is the result produced by it.

We consider the use of feature extraction for coping with the curse of dimensionality in the dynamic integration of classifiers [Tsymbal, 2003]. The FEDIC (Feature Extraction for Dynamic Integration of Classifiers) algorithm was proposed which combines the dynamic selection and dynamic voting classifier integration techniques with the conventional PCA and two supervised eigenvector-based approaches that use the within- and between-class covariance matrices.

Our main hypothesis has been that with the data sets for which feature extraction improves classification accuracy employing a base classifier (such as *kNN* or Naïve Bayes), it will also improve classification accuracy when employing a dynamic integration approach. Conversely, we expected that with data sets for which feature extraction decreases or has no effect on classification accuracy with the base classifier, it will also decrease or will have no effect on classification accuracy employing a dynamic integration approach. This hypothesis was supported by the results obtained during the experiments conducted on a number of data sets from the UCI machine learning repository.

Decision Support System for the Best-suited Technique and Its Parameters Selection

Summarising the results of the up to this date research in the area we can state that there is no feature extraction technique that would be the best for any data set given with respect to the task of classification. Thus the problem of adaptive selection of the most suitable feature extraction technique for a data set needs further research work. We do not have canonical knowledge, perfect mathematical models or any relevant tool to select the best-suited technique. Thus, we are dealing with so-called empirical domain area having a volume of accumulated empirical facts, some trends and some dependencies found. And the theoretical summarization of these facts, trends and dependencies is the question of future research.

These prerequisites lead us on to consider the possibility of decision support system developing based on the methodology of expert system design in order to help to manage the data mining process with regard to the selection of the best-suited combination for a classification task. The main goal of such a system is to recommend the best-suited feature extraction method and a classifier for a given data set according to a set of rules related to a given problem. Achieving this goal produces a great benefit in the sense that it would be possible to come from the *wrapper* type approach to the *filter* paradigm [Hall, 2000]. In the wrapper type approach the interaction between the feature selection process and the construction of the classification model is assumed and the parameter tuning for every stage and for every method is needed. In the filter paradigm evaluation process is independent from the learning algorithm and the methods, and their parameters' selection process is performed according to a certain set of criteria before the algorithm starts. However, an additional goal of the prediction of model's output performance needs also further consideration.

The coverage of the responsibilities of the decision support system in the data mining process is depicted on fig. 1 (left). It can be seen that as soon as a training data set comes to the data mining system and the preliminary data preparation and data cleaning processes are finished, the Decision Support System takes responsibility to manage the processes of feature extraction and classification, namely to select the best-suited methods and the best parameters for those methods. And only after the model is built and validated, it comes to final evaluation on a test set.

The basic structure of the DSS is presented on fig.1 (right). The "heart" of this system is the *Knowledge Base* (KB) that contains a set of facts about the domain area and a set of rules in a symbolic form describing the logical references between the "symptoms" that are a concrete classification problem and recommendations about the best-suited model for a given problem. These facts and rules can be a basis for the new ones. Generally, the knowledge base is a dynamic part of the system that can be supplemented and updated through the knowledge acquisition and knowledge refinement processes. The content of the knowledge base is updated when time elapse, and, in some cases, even during solving the task.

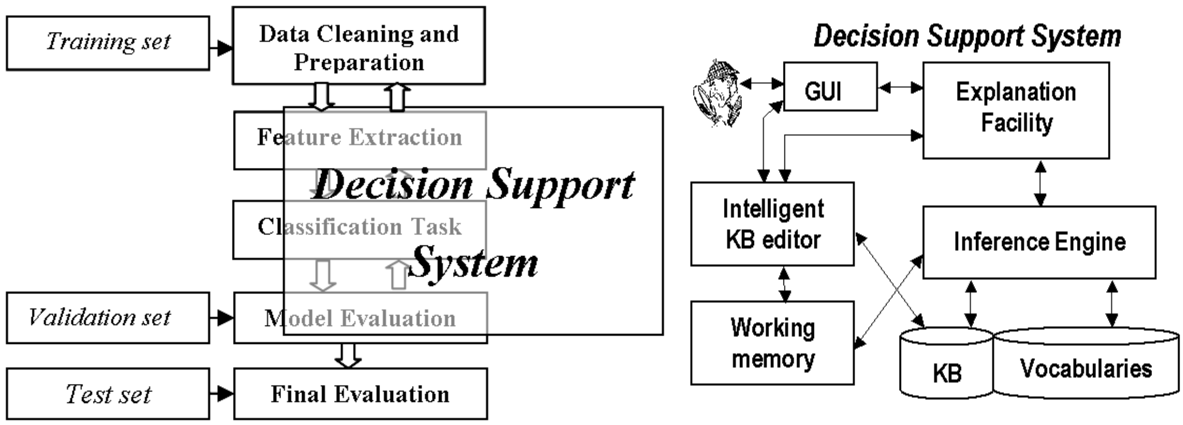

Figure 1 – Integration of the decision support system (right) into the data mining process (left).

Vocabularies contain the lists of terms that include feature extraction methods and their input parameters, classifiers and their input and output parameters, and three types of data set characteristics: simple measures such as the number of instances, the number of attributes, and the number of classes; statistical measures such as the departure from normality, correlation within attributes, the proportion of total variation explained by the first *k* canonical discriminants; and information-theoretic measures such as noisiness of attributes, the number of irrelevant attributes, and the mutual information of class and attribute.

Inference Engine is the “brain” of the system. It can be considered as a rule interpreter. It is a logical programming component of the system that realises a reasoning mechanism based on the Knowledge Base and the Working Memory information. This subsystem is the backbone of the consultation process since it produces the information how the system comes to a conclusion. The inference engine is able to search for the missing knowledge either by asking from a researcher or by conducting additional experiments if there is not enough information to come to a confident conclusion. This subsystem should contain at least three main components: an interpreter that executes some agenda: a set of rules from the Knowledge Base; a scheduler that controls the execution of the agenda; a consistency enforcer that maintains the reasoning process used to obtaine the decisions.

The Working Memory contains all the data that is essential for the current problem including input data, results of inference, and intermediate results.

The Intelligent KB Editor is a tool that aims to provide intelligent methodologies for refinement of knowledge accumulated in the knowledge base and insertion of new knowledge got from the knowledge acquisition process. Particularly, the Editor should include the patterns of knowledge representation language and provide a tool for a knowledge engineer to develop the knowledge base in a dialogue mode. Beside these common functions related to the Editor, its tasks include experimentation routing management that will be considered in the next section.

The Explanation Facility is the subsystem that is aimed to trace problem solving steps to the obtaining results. Intermediate conclusions and the consequence of applied rules are stored in the tree of conclusion. Successful application of a rule corresponds to moving to another node of the tree during reaching a goal statement. The explanation system should be able to answer on the questions as: how was a certain conclusion inferred? why was additional information requested? how was the output model’s performance estimated?

The Graphical User Interface (GUI) provides an interactive environment that links the KB, Intelligent KB Editor and knowledge engineer as well as the Explanation Facility module with the users of the system.

Filling the knowledge base is among the most challenging task related to the development of the DSS and it will be in the heart of our research focus. Potential contribution might be found discovering a number of criteria from the experiments conducted on artificially generated data sets with pre-defined characteristics examining the dependencies between the characteristics of a data set in general and the characteristics of every local partition of the instance space in particular, and the type and parameters of the feature extraction approach best suited for the data set will help to define a set of criteria that can be applied for the generation of rules needed for the decision-making system.

The Knowledge Acquisition Process

The Knowledge Base is a dynamic part of the system that can be supplemented and refreshed through The Intelligent KB Editor. We should notice that there are two potential sources of knowledge to be discovered for the proposed system. These are the analysis of theory background that lies behind the feature extraction and classification methods, and field experiments.

In the first case, knowledge is formulated by an expert in the area of the specific feature extraction methods and classification schemes, and then represented as a set of rules by a knowledge engineer in the terms of a knowledge representation language that is supported by the system. We argue that it is possible and reasonable to categorise the facts and rules that are present in the Knowledge Base. Categorisation can be done according to the way the knowledge has been obtained – has it been got from the analysis of experimental results of from the domain theory, was it put automatically by the Intelligent KB Editor or by a knowledge engineer (who could be a data miner as well). Another categorisation criterion is the level of confidence of a rule. The expert can be sure in a certain fact or may just think or to hypothesize about another fact. In a similar way, a rule that has been just generated from the analysis of results by experimenting on artificially generated data sets but has been never verified on real-worlds data sets and a rule that has been verified on a number of real-world problems. These two rules definitely should not have the same level of confidence.

In addition to the “trust“ criteria due to the categorisation of the rules it is possible to adapt the system to a concrete researcher needs and preferences by giving higher weights to the rules that actually are the ones of the user.

And, in the second case, a data miner can discover knowledge during the analysis of results obtained from the experiments as separate facts, trends and dependencies. In the same manner, discovered knowledge is represented as a set of rules by a knowledge engineer using of the knowledge representation language. Alternatively, the knowledge acquisition process can be automatic, i.e. the knowledge discovery process would be accomplished without any interference with a human expert. This may happen using the possibility of deriving new rules and updating the old ones based on the analysis of results obtained during the self-run experimenting.

In both the last cases we have a problem of learning how the Intelligent KB Editor should try to build up a classification or a regression model on meta-data resulted from experiments. In this context the input parameters for a classification model are specific data set characteristics and a classification model’s outputs that include accuracy, sensitivity, specificity, time complexity, etc. The combination of a feature extraction method’s and a classification model’s names with their parameter values represents a class label. When building a regression model – meta-data-set attributes are data set characteristics, the feature extraction method’s and the classification model’s names, and one of the model output characteristics is the attribute which value (continuous) has to be predicted.

Then, in terms of attribute-value (feature-value) notation, each instance can be represented in the following way:

$$\mathbf{x} = [v(x_{DS_1}), \ldots, v(x_{DS_l}), v(x_{MO_1}), \ldots, v(x_{MOm_1})],$$

where $v(x_{DSi})$ denotes the value of attribute x_{DSi} that represents one of the data set characteristics, and $v(x_{MOi})$ denotes the value of attribute x_{MOi} that represents one of the model output characteristics, and $l + m = p$ is the number of attributes that constitute the meta-data-set.

And placing an instance into one of a finite set of possible categories can be depicted as

$$C(\mathbf{x}) \in range(\mathbf{y}),$$

where $range(\mathbf{y})$ denotes the set of possible values for the categorical output attribute, class value *y*. In our case class value is assigned to every distinct combination of a feature extraction method and a classifier with their parameters values.

The results obtained by us up to the present stage of research show a high level of complexity in dependencies between the data set characteristics and the best-suited scheme for the data mining process. In order to further develop our understanding it is necessary to proceed the research with the following iterations:

- Generation of artificial data sets with known characteristics (simple, statistical and information-theoretic measures);
- Design of experiments on the generated artificial data sets;
- Derivation of dependencies and definition of the criteria from the obtained results;
- Development of a knowledge base defining a set of rules on the set of obtained criteria;
- Proof of the constructed theory with a set of experiments on real-world data sets.

Thus, three basic research methods are used in the research: the theoretical approach, the constructive approach, and the experimental approach. These approaches are closely related and are applied in parallel. The theoretical backgrounds are exploited during the constructive work and the constructions are used for experimentation. The results of constructive and experimental work are used to refine the theory.

An example of such a procedure can be presented as:

- Generation of artificial data sets with the number of attributes from 2 to 100, with the number of instances from 150 to 5000, with the number of classes from 2 to 10, with the average correlation between the attributes from 10% to 90%, with the average noisiness of attributes from 10% to 50%, with the percent of irrelevant attributes from the total number of attributes from 10% to 50%.
- Design of the experiments on generated artificial data sets and analysing accuracy and efficiency of the classification model built on different learning algorithms and using different feature extraction methods. Tuning of the input parameters for each combination is required.
- Analysis of the dependencies and trends between output accuracies and efficiencies, feature extraction methods and classifiers, their input parameters, and pre-defined data set characteristics.
- Definition of a set of rules that reflect found dependencies and trends.
- Execution of a number of experiments on UCI data sets using DSS for the best-suited feature extraction method and classifier selection.
- Addition of the invented rules that were successfully validated during the tests on the benchmark data sets to the knowledge base.

Conducting a number of experiments on artificial data sets with pre-defined characteristics, according to the example shown above, we will get an input space x, i.e. as the one presented in Table 1.

Table 1 – An example of the hypothetic meta-data-set.

Data set characteristics						Model output characteristics				ModelID (class label)
Simple		Statistical		Inf. Theoretic		Accuracy		Complexity		
attributes	classes	corr.	normality	noise	entropy	accuracy	diversity	training time	test time	
$v(x_{DS_1}^{(1)})$			…		$v(x_{DS_l}^{(1)})$	$v(x_{MO_1}^{(1)})$	…		$v(x_{MOm}^{(1)})$	C(x1)
…					…	…			…	…
$v(x_{DS_1}^{(n)})$			…		$v(x_{DS_l}^{(n)})$	$v(x_{MO_1}^{(n)})$	…		$v(x_{MO_{m1}}^{(n)})$	C(xn)

We consider a decision tree learning algorithm as a mean of automatic rule extraction for the knowledge base. Decision tree learning is one of the most widely used inductive learning methods [Quinlan, 1993]. A decision tree is represented as a set of nodes and arcs. Each node contains a feature (an attribute) and each arc leaving the node is labelled with a particular value (or range of values) for that feature.

Together, a node and the arcs leaving it represent a decision about the path an example follows when being classified by the tree. Given a set of training examples, a decision tree is induced in a “top-down” fashion by repeatedly dividing up the examples according to their values for a particular feature. This is known as a “divide and conquer” or “recursive partitioning” approach to learning. Initially all the examples are in one partition and each feature is evaluated for its ability to improve the “purity” of the classes in the partitions it produces. The splitting process continues recursively until all of the leaf nodes are of one class.

At the Figure 2 an example of the part of the abstract model built by decision tree on the meta-training set is presented. By means of analysing the tree branches it is possible to generate “if-then” rules for the knowledge base. A rule reflects certain relationship between meta-data-set characteristics and a combination of a feature extraction method and a classification model.

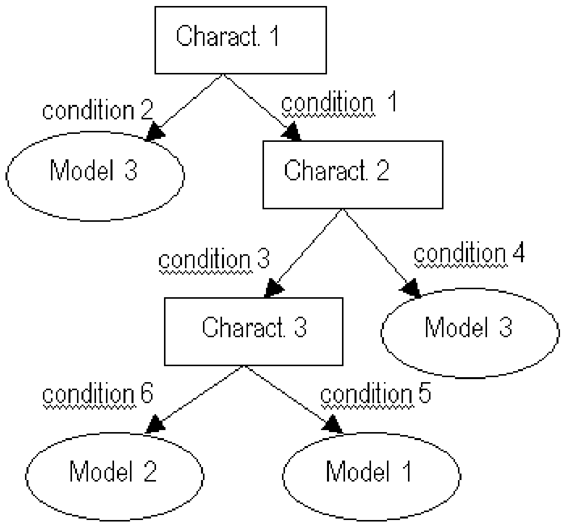

Figure 2 – The example of the part of abstract model built by decision tree.

Conclusion

Feature extraction is one of the dimensionality reduction techniques that are often used to struggle against the problems caused by the “curse of dimensionality”. In this paper we considered three eigenvector-based feature extraction approaches that were applied for different classification problems. We presented the summary of results that shows a high level of complexity in dependencies between the data set characteristics and the best-suited scheme for the data mining process. There is no feature extraction method that would be the most suitable for all classification tasks. Due to the fact that there is no well-grounded strong theory that would help to build up an automated system for such feature extraction method selection, a decision support system that would accumulate separate facts, trends and dependencies between the data characteristics and output parameters of classification schemes performed in the spaces of extracted features was proposed.

We considered the goals of such a system, the basic ideas that define its structure and methodology of knowledge acquisition and validation. The Knowledge Base is the basis for the intellectuality of the expert system. That is why we recognised the problem of discovering rules from the experiments of an artificially generated data set with known predefined simple, statistical and information-theoretic measures, and validation of those rules on benchmark data sets as a prior research focus in this area.

It should be noticed that generally the proposed approach has a serious limitation. Namely the drawbacks can be expressed in the terms of fragmentariness and incoherence (disconnectedness) of the components of knowledge to be produced. And we definitely do not claim about completeness of our decision support system. Otherwise, certain constrains and assumptions to the domain area were considered, and limited sets of feature extraction methods, classifiers and data set characteristics were considered in order to guarantee the desired level of confidence in the system when solving a bounded set of problems.

Acknowledgements

This research is partly supported by the COMAS Graduate School of the University of Jyväskylä, Finland and by Science Foundation Ireland. We would like to thank the UCI ML repository of databases, domain theories and data generators for the data sets, and the MLC++ library [Kohavi, 1996] for the source code used in our studies.

Bibliography

[Aivazyan, 1989] Aivazyan, S.A.: Applied Statistics: Classification and Dimension Reduction. Finance and Statistics, Moscow, 1989.

[Bellman, 1961] Bellman, R., Adaptive Control Processes: A Guided Tour, Princeton University Press, 1961.

[Blake, 1998] Blake, C.L., Merz, C.J. UCI Repository of Machine Learning Databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Dept. of Information and Computer Science, University of California, Irvine CA, 1998.

[Fayyad, 1996a] Fayyad U.M., Piatetsky-Shapiro G., Smyth P., and Uthurusamy. Advances in Knowledge Diskovery and Data Mining. *AAAI Press/MIT Press*, 1996

[Fayyad, 1996b] Fayyad U.M. Data Mining and Knowledge Discovery: Making Sense Out of Data, *IEEE Expert*, Vol. 11, No. 5, Oct., 1996, pp. 20-25

[Fukunaga, 1991] Fukunaga, K. Introduction to Statistical Pattern Recognition. Academic Press, London, 1991.

[Hall, 2000] Hall, M.A. Correlation-based feature selection of discrete and numeric class machine learning. In Proc. Int. Conf. On Machine Learning (ICML-2000), San Francisco, CA. Morgan Kaufmann, San Francisco, CA, 359-366, 2000.

[Jackson, 1999] Jackson P. Introduction to Expert Systems, 3rd Edn. Harlow, England: Addison Wesley Longman, 1999.

[Jolliffe, 1986] Jolliffe, I.T. Principal Component Analysis. Springer, New York, NY. 1986.

[Kohavi, 1996] Kohavi, R., Sommerfield, D., Dougherty, J. Data mining using MLC++: a machine learning library in C++. Tools with Artificial Intelligence, IEEE CS Press, 234-245, 1996.

[Liu, 1998] Liu H. Feature Extraction, Construction and Selection: A Data Mining Perspective, ISBN 0-7923-8196-3, Kluwer Academic Publishers, 1998

[Merz, 1996] Merz, C.: Dynamical Selection of Learning Algorithms. In: D.Fisher, H.-J.Lenz (eds.), Learning from Data, Artificial Intelligence and Statistics, Springer-Verlag, NY, 1996.

[Oza, 1999] Oza, N.C., Tumer, K. Dimensionality Reduction Through Classifier Ensembles. Technical Report NASA-ARC-IC-1999-124, Computational Sciences Division, NASA Ames Research Center, Moffett Field, CA, 1999.

[Quinlan, 1993] Quinlan, J.R. 1993. C4.5 Programs for Machine Learning. San Mateo CA: Morgan Kaufmann.

[Tsymbal, 2002] Tsymbal A., Puuronen S., Pechenizkiy M., Baumgarten M., Patterson D. Eigenvector-based feature extraction for classification, In: *Proc. 15th Int. FLAIRS Conference on Artificial Intelligence*, Pensacola, FL, USA, AAAI Press, 354-358, 2002.

[Tsymbal, 2003] Tsymbal A., Pechenizkiy M., Puuronen S., Patterson D.W., Dynamic integration of classifiers in the space of principal components, In: L.Kalinichenko,R.Manthey, B.Thalheim, U.Wloka (eds.), Proc. Advances in Databases and Information Systems: 7th East-European Conf. ADBIS'03, Dresden, Germany,Lecture Notes in Computer Science, Vol. 2798, Springer-Verlag, 2003,278-292.

[Turban, 2001] Turban E. and Aronson J.E. Decision Support Systems and Intelligent Systems, Prentice-Hall, 2001.

Author information

Mykola Pechenizkiy - Department of Computer Science and Information Systems, University of Jyväskylä, P.O.Box: 35, Jyväskylä 40351, Finland; e-mail: mpechen@cs.jyu.fi

Seppo Puuronen – Department of Computer Science and Information Systems, University of Jyväskylä, P.O.Box: 35, Jyväskylä 40351, Finland; e-mail: sepi@cs.jyu.fi

Alexey Tsymbal - Department of Computer Science, Trinity College Dublin, College Green, Dublin 2, Ireland;
e-mail: Alexey.Tsymbal@cs.tcd.ie