JYVÄSKYLÄ STUDIES IN COMPUTING
56

Mykola Pechenizkiy

Feature Extraction for Supervised Learning in Knowledge Discovery Systems

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 56

Mykola Pechenizkiy

Feature Extraction for Supervised Learning in Knowledge Discovery Systems

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksessa (Ag Aud. 2)
joulukuun 20. päivänä 2005 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in the building Agora, Ag Aud. 2, on December 20, 2005 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2005

Feature Extraction for Supervised Learning in Knowledge Discovery Systems

JYVÄSKYLÄ STUDIES IN COMPUTING 56

Mykola Pechenizkiy

Feature Extraction for Supervised Learning in Knowledge Discovery Systems

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2005

Editors
Seppo Puuronen
Department of Computer Science and Information Systems, University of Jyväskylä
Pekka Olsbo, Marja-Leena Tynkkynen
Publishing Unit, University Library of Jyväskylä

URN:ISBN:9513922715
ISBN 951-39-2271-5 (PDF)

ISBN 951-39-2299-5 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2005

ABSTRACT

Pechenizkiy, Mykola
Feature Extraction for Supervised Learning in Knowledge Discovery Systems
Jyväskylä: University of Jyväskylä, 2005, 86 p. (+ included articles)
(Jyväskylä Studies in Computing,
ISSN 1456-5390; 56)
ISBN 951-39-2271-5
Finnish summary
Diss.

Knowledge discovery or data mining is the process of finding previously unknown and potentially interesting patterns and relations in large databases. The so-called "curse of dimensionality" pertinent to many learning algorithms, denotes the drastic increase in computational complexity and classification error with data having a great number of dimensions. Beside this problem, some individual features, being irrelevant or indirectly relevant for the learning concepts, form poor problem representation space. The purpose of this study is to develop theoretical background and practical aspects of feature extraction (FE) as means of (1) dimensionality reduction, and (2) representation space improvement, for supervised learning (SL) in knowledge discovery systems. The focus is on applying conventional Principal Component Analysis (PCA) and two class-conditional approaches for two targets: (1) for a base level classifier construction, and (2) for dynamic integration of the base level classifiers. Theoretical bases are derived from classical studies in data mining, machine learning and pattern recognition. The software prototype for the experimental study is built within WEKA open-source machine-learning library in Java. The different aspects of the experimental study on a number of benchmark and real-world data sets include analyses of (1) importance of class information use in the FE process; (2) (dis-)advantages of using either extracted features or both original and extracted features for SL; (3) applying FE globally to the whole data and locally within natural clusters; (4) the effect of sampling reduction on FE for SL; and (5) the problems of FE techniques selection for SL for a problem at consideration. The hypothesis and detailed results of the many-sided experimental research process are reported in the corresponding papers included in the thesis. The main contributions of the thesis can be divided into contribution (1) to current theoretical knowledge and (2) to development of practical suggestion on applying FE for SL.

Keywords: feature extraction, dimensionality reduction, principal component analysis, data pre-processing, integration of data mining methods, supervised learning, knowledge discovery in databases

ACM Computing Reviews Categories

H.2.8	Information Systems: Database Management: *Database Applications, Data Mining*
I.2.6	Computing Methodologies: Artificial Intelligence: *Learning*
I.5.1	Computing Methodologies: Pattern Recognition: *Models*
I.5.2	Computing Methodologies: Pattern Recognition: *Design Methodology*

Author's address

Mykola Pechenizkiy
Dept. of Computer Science and Information Systems
University of Jyväskylä
P. O. Box 35, 40351 Jyväskylä, Finland
E-mail: Mykola.Pechenizkiy@it.jyu.fi

Supervisors

Prof. Dr. Seppo Puuronen
Dept. of Computer Science and Information Systems
University of Jyväskylä
Finland

Dr. Alexey Tsymbal
Department of Computer Science
Trinity College Dublin
Ireland

Prof. Dr. Tommi Kärkkäinen
Department of Mathematical Information Technology
University of Jyväskylä
Finland

Reviewers

Prof. Dr. Ryszard Michalski
Machine Learning and Inference Laboratory
George Masson University
USA

Prof. Dr. Peter Kokol
Department of Computer Science
University of Maribor
Slovenia

Opponent

Dr. Kari Torkkola
Motorola Labs
USA

ACKNOWLEDGEMENTS

I would like to thank all those who helped me to carry out this research. I am thankful to my chief scientific supervisor Prof. Seppo Puuronen (Dept. of Computer Science and Information Systems, University of Jyväskylä), who never refused to share his time and scientific experience on the direction of the thesis in many fruitful discussions.

I am very grateful to Dr. Alexey Tsymbal who has been able to combine successfully of being my second scientific supervisor, colleague and friend since 2001. He has introduced many new things to me and helped me to become a researcher. His doctoral thesis "*Dynamic Integration of Data Mining Methods in Knowledge Discovery Systems*" was of great help for me in enabling me to better understand some concepts and to present my thesis as a collection of papers with a summary.

I would also like to thank Prof. Seppo Puuronen, Dr. Alexey Tsymbal and Dr. David Patterson for their co-operation and co-authoring of the articles included in this thesis.

I am thankful to my third scientific supervisor Prof. Tommi Kärkkäinen (Dept. of Mathematical Information Technology, University of Jyväskylä) for his valuable advices and comments. My special thanks to Prof. Tommi Kärkkäinen and Licentiate Pekka Räsänen for their supervision of my M.Sc. project "*PCA-based Feature Extraction for Classification*". That work was the starting point of the present doctoral thesis.

Prof. Ryszard Michalski from George Masson University and Prof. Peter Kokol from University of Maribor have acted as external reviewers (examiners) of the dissertation. Their constructive comments and suggestions helped me to improve this work.

I would like to thank Dr. David Patterson who invited me to Northern Ireland Knowledge Engineering Laboratory, University of Ulster (1.09 – 30.11.2002), as a visiting researcher, and Prof. Padraig Cunningham who invited me to Machine Learning Group, Trinity College Dublin (1.04 – 30.04.2003 and 17.07 – 7.08.2005). This cooperation gave me useful experience and generated new ideas for further research.

This research took place at the Department of Computer Science and Information Systems, at University of Jyväskylä and I am thankful to many staff members for their support. I am also thankful to the Department for their financial support of conference traveling.

This research has been funded by the Graduate School in Computing and Mathematical Sciences (COMAS) of the University of Jyväskylä. I am also grateful to the INFWEST.IT program for some practical arrangements.

I would also like to thank Steve Legrand for his support in checking the language.

I would also like to thank the UCI machine learning repository of databases, domain theories and data generators (UCI ML Repository) for the datasets, and the machine learning library in C++ (MLC++) and WEKA developers for the source code used in this research.

Last, but not least, I wish to thank my wife Ekaterina, my dear parents Anatoliy and Lyubov, my brother Oleksandr, and all my friends, who provided invaluable moral support.

Jyväskylä
November 2005

Mykola Pechenizkiy

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

C4.5	C4.5 decision tree classifier
DIC	dynamic integration of classifiers
DB	data base
DM	data mining
DS	dynamic selection
DV	dynamic voting
DVS	dynamic voting with selection
DSS	decision support system
FE	feature extraction
FS	feature selection
FEDIC	feature extraction for dynamic integration of classifiers
IS	information system
ISD	information system development
KDD	knowledge discovery in databases
KDS	knowledge discovery system
kNN	(k) nearest neighbour algorithm
LDA	linear discriminant analysis
ML	machine learning
MLC++	machine learning library in C++
OLE DB	Object Linking and Embedding DB
PCA	principal component analysis
RP	random projection
SL	supervised learning
WEKA	Waikato Environment for Knowledge Analysis
WNN	a weighted average of the nearest neighbours

CONTENTS

LIST OF INCLUDED ARTICLES

I Tsymbal, A., Puuronen, S., Pechenizkiy, M., Baumgarten, M. & Patterson, D. 2002. Eigenvector-based Feature Extraction for Classification. In: S. Haller, G. Simmons (Eds.), Proceedings of 15th International FLAIRS Conference on Artificial Intelligence, FL, USA: AAAI Press, 354-358.

II Pechenizkiy, M., Tsymbal, A. & Puuronen, S. 2004. PCA-based Feature Transformation for Classification: Issues in Medical Diagnostics. In: R. Long *et al.* (Eds.), Proceedings of 17th IEEE Symposium on Computer-Based Medical Systems CBMS'2004, Bethesda, MD: IEEE CS Press, 535-540.

III Pechenizkiy, M., Tsymbal, A. & Puuronen, S. 2005. On Combining Principal Components with Fisher's Linear Discriminants for Supervised Learning. (submitted to) Special Issue of Foundations of Computing and Decision Sciences "Data Mining and Knowledge Discovery" (as extended version of Pechenizkiy *et al.*, 2005e).

IV Pechenizkiy, M. 2005. The Impact of the Feature Extraction on the Performance of a Classifier: kNN, Naïve Bayes and C4.5. In: B. Kegl & G. Lapalme (Eds.), Proceedings of 18th CSCSI Conference on Artificial Intelligence AI'05, LNAI 3501, Heidelberg: Springer Verlag, 268-279.

V Pechenizkiy, M., Tsymbal, A. & Puuronen, S. 2005. Supervised Learning and Local Dimensionality Reduction within Natural Clusters: Biomedical Data Analysis, (submitted to) IEEE Transactions on Information Technology in Biomedicine, Special Post-conference Issue "Mining Biomedical Data" (as extended version of Pechenizkiy *et al.*, 2005c).

VI Pechenizkiy, M., Puuronen, S. & Tsymbal, A. 2006. The Impact of Sample Reduction on PCA-based Feature Extraction for Supervised Learning. (to appear) In: H. Haddad *et al.* (Eds.), Proceedings of 21st ACM Symposium on Applied Computing (SAC'06, Data Mining Track), ACM Press.

VII Pechenizkiy, M., Tsymbal, A., Puuronen, S. & Patterson D. 2005. Feature Extraction for Dynamic Integration of Classifiers, (submitted to) Fundamenta Informaticae, IOS Press (as extended version of Tsymbal *et al.*, 2003).

VIII Pechenizkiy, M., Puuronen, S. & Tsymbal, A. 2003. Feature Extraction for Classification in Knowledge Discovery Systems. In: V. Palade, R. Howlett & L. Jain (Eds.), Proceedings of 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems KES'2003, LNAI 2773, Heidelberg: Springer-Verlag, 526-532.

IX Pechenizkiy, M. 2005. Data Mining Strategy Selection via Empirical and Constructive Induction. In: M. Hamza (Ed.), Proceedings of the IASTED International Conference on Databases and Applications DBA'05, Calgary: ACTA Press, 59-64.

1 INTRODUCTION

Knowledge discovery in databases (KDD) or data mining (DM) is the process of finding previously unknown and potentially interesting patterns and relations in large databases (Fayyad 1996). Numerous data mining methods have recently been developed to extract knowledge from these large databases. Selection of the most appropriate data mining method or a group of the most appropriate methods is usually not straightforward.

During the past several years in a variety of application domains researchers have tried to learn how to manage knowledge discovery process in those specific domains. This has resulted in a large number of “vertical solutions”. 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. These systems should be able to discover knowledge by combining several available techniques, and provide a more automatic environment, or an application envelope, surrounding a highly sophisticated data mining engine (Fayyad & Uthurusamy 2002).

This thesis presents the study of data mining techniques integration and application for different benchmark and real-world data sets, with the focus on the study of feature extraction (FE) for supervised learning (SL). This introductory chapter presents the motivation for the research and the main research objectives, overviews the methods used in the study, and introduces the organization of the thesis.

1.1 Motivation

Classification is a typical data mining task where the value of some attribute for a new instance is predicted based on the given collection of instances for which all the attribute values are known (Aivazyan, 1989). The purpose of supervised learning (SL) is to *learn to classify* (predict a value of some attribute for) a new instance. 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" (Bellman, 1961) pertinent to many learning algorithms, denotes the drastic increase in computational complexity and classification error with data having a great number of dimensions (Aivazyan, 1989). Furthermore, nowadays the complexity of real-world problems, increased by the presence of many irrelevant or indirectly relevant features, challenge the existing learning algorithms. It is commonly accepted that just by pushing a button someone should not expect useful results to appear.

Hence, attempts are often made to reduce the dimensionality of the feature space before SL is undertaken. Feature extraction (FE) is one of the dimensionality reduction techniques that 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 (Liu, 1998).

FE is an effective data pre-processing step aimed to reduce the dimensionality and to improve representation space of the problem at consideration.

Although there has been some rigorous research going on both FE and SL for many years in applied statistics, pattern recognition and related fields, to the best of our knowledge, there is no comprehensive many-sided analysis of FE and SL processes integration.

1.2 Objectives

In this thesis, the emphasis is on studying FE and SL integration. Both FE and SL are seen as constituent parts of a DM strategy. Our basic assumption is that each DM strategy is best suited for a certain problem. Therefore, our overall (faraway) research goal is to contribute to knowledge in the problem of DM strategy selection for a certain DM problem. And our particular focus is on different combinations of FE techniques and SL techniques.

The main objective of this study is to develop theoretical background and practical aspects of FE as means of (1) dimensionality reduction and (2) representation space improvement for SL in knowledge discovery systems. The focus is on applying conventional Principal Component Analysis (PCA) and two class-conditional approaches for two targets: (1) for a base level classifier construction, and (2) for dynamic integration of the base level classifiers. The different aspects of the study include analyses of (1) importance of class information use in the FE process; (2) advantages of using either extracted features or both original and extracted features for SL; (3) applying FE globally to the whole set of training instances and locally within natural clusters; and (4) the effect of sample reduction on FE for SL. Besides this, more general problem of FE techniques selection for SL for a dataset at consideration is analysed.

Related work on FE for SL includes also research on constructive induction (Michalski, 1997) and latent semantic indexing for text mining applications (Deerwester *et al.*, 1990).

1.3 Methods and Results

We consider a knowledge discovery system as a special kind of adaptive information system. We adapted the Information System Development (ISD) framework for the context of DM systems development. Three basic groups of IS research methods, including conceptual-theoretical, constructive, and experimental approaches are used in this study. These approaches are tightly connected and are applied in parallel. The theoretical background is 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.

Consequently, the main results (beside the developed software prototype for experimental studies) come from the experimental study.

The results of our study show that:

- FE is an important step in DM/KDD process that can be beneficial for SL and integration of classifiers in terms of classification accuracy and in terms of time complexity of model learning and new instances classification.
- FE can improve classification accuracy of a model produced by a learner even for datasets having relatively small number of features. And, therefore, FE can be considered as a dimensionality reduction technique as well as a technique for construction of better representation space for further supervised learning.
- Use of class information in FE process is crucial for many datasets.
- Combination of original features with extracted features can be beneficial for SL on some datasets.
- Local FE and SL models can outperform corresponding global models in classification accuracy using fewer of features for learning.
- Training sample reduction affects the performance of the SL with FE rather differently; and when the proportion of training instances used to build the FE and the learning model is relatively small it is important to use an adequate sample reduction technique to select more representative instances for the FE process.

In our experimental study we use mainly benchmark data sets from UCI repository (Blake & Merz, 1998). However, part of the experimental study was done on data related to some real problems of medical diagnostics (the classification of acute abdominal pain (Zorman *et al.*, 2001) and problems of antibiotic resistance (Pechenizkiy *et al.*, 2005h)). Besides benchmark and real-world data sets we conduct some studies on synthetically generated data sets where desired data set characteristics are varied.

1.4 Thesis overview

The thesis consists of two parts. The first part presents the summary of the collection of papers presented in the second part (the included papers are listed before this introductory Chapter). The summary part of the thesis introduces the background of the study, presents the research problem of the study, describes basic research methods used, overviews the papers included in the thesis and concludes with the main contribution of the thesis and suggestions for further research directions.

The organization of the thesis summary part is as follows: in Chapter 2 research background is considered. First, in Section 2.1 knowledge discovery and data mining concepts are discussed. A brief history of knowledge discovery systems is presented. Then, basic introduction to the problem of classification (Section 2.2), to dynamic integration of classifiers (Section 2.3), to dimensionality reduction (Section 2.4), to FE for supervised learning (Section 2.5) that is the focus of this thesis, and to selection of representative instances for FE (Section 2.6) is presented. In Chapter 3, the research problem of the thesis is stated. Each aspect of the study is presented with a separate section. Chapter 4 introduces the research design of the thesis. First, research methods being used and basic approaches for evaluating learned models that are used in experiments are discussed. Then, experimental design is considered. Chapter 5 contains summaries of the articles included in the thesis. Each section is a summary of the corresponding included article. In Chapter 6, contribution of the thesis is summarized, and limitations of the research and future work are discussed. The information about datasets used in the experimental studies is given in Appendix A.

2 RESEARCH BACKGROUND

Knowledge discovery in databases (KDD) is a combination of data warehousing, decision support, and data mining – an innovative new approach to information management (Fayyad, 1996). KDD is an emerging area that covers such areas as statistics, machine learning, databases, pattern recognition, econometrics, and some other. In the following section we consider knowledge discovery as a process and discuss the perspectives in KDD systems. In the further section, basics of supervised learning are introduced and classifiers used in the study are considered, and some background on ensemble classification and dynamic integration of classifiers is presented. Then, we introduce the problem known as "the curse of dimensionality", before feature extraction techniques for supervised learning are considered. We finish the chapter with a review of training sample selection techniques (used in this study) aimed to reduce the computational complexity of feature extraction and supervised learning.

2.1 Knowledge discovery in databases

The present history of KDD systems development consists of three main stages/generations (Piatetsky-Shapiro, 2000). The year 1989 can be considered as the first generation of KDD systems when few single-task data mining (DM) tools such as C4.5 decision tree algorithm (Quinlan, 1993) existed. These tools were difficult to use and required significant preparation. Most of such systems were based on a loosely coupled architecture, where the database and the data mining subsystems are realised as separate independent parts. This type of architecture demands continuous context switching between the data mining engine and the database (Imielinski & Mannila, 1996).

The year 1995 can be associated with a formation of the second-generation tool-suits (Piatetsky-Shapiro, 2000). KDD started to be seen as "the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad 1996, 22).

2.1.1 Knowledge discovery as a process

The process of KDD comprises several steps, which involve data selection, data pre-processing, data transformation, application of machine learning (ML) techniques (Mitchel, 1997), and interpretation and evaluation of patterns. These basic steps of the KDD process from raw data to the extracted knowledge are presented in Figure 1 (adapted from (Fayyad, 1996)).

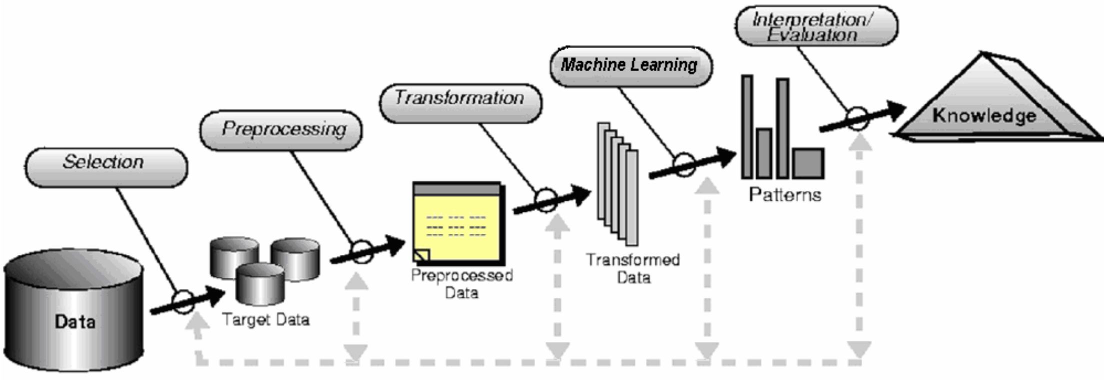


FIGURE 1 Basic steps of KDD process (adapted from (Fayyad 1996, 22)). Solid arrows denote the processing steps of data towards discovered knowledge, while dotted-line arrows show that each of these steps may form a different iterative cycle and results of one step can be used for any other step.

The process starts from target data selection that is often related to the problem of building and maintaining useful data warehouses (Fayyad & Uthurusamy, 2002) and is recognized as the most time-consuming in the KDD process (Fayyad, 1996). The target data selection is excluded from our research focus here. After selection, the target data is preprocessed in order to reduce the level of noise, to preprocess the missing information, to reduce data, and to remove obviously redundant features.

The data transformation step is aimed either to reduce the dimensionality of the problem, extracting the most informative features, or to enlarge feature space, constructing additional (potentially useful) features. The linear FE techniques (a type of techniques that can be applied during the transformation step) for subsequent classification, is, in fact, the main focus of this thesis. During the next step, search for patterns – summary of a subset of the data, statistical or predictive models of the data, and relationships among parts of the data – i.e., application of machine-learning techniques takes place. In Fayyad

(1996) this step is associated with data mining – the identification of interesting structure in the data. In this thesis we prefer to denote the step associated with the search for patterns as application of ML techniques. Data mining concept is used when we consider the feature extraction/transformation processes, application of ML techniques, and evaluation processes as a core process of KDD. These processes are in the focus of this thesis, namely studying of FE and classification processes, their integration and evaluation. When we refer here to data mining strategy selection, we assume a selection of the most appropriate FE technique (as one type of techniques for data transformation), classifier and evaluator.

The Interpretation/evaluation step helps the user by providing tools for visualization (Fayyad *et al.*, 2001) of models built and patterns discovered and for generation of reports with discovery results and discovery process log analysis. The user has a possibility to interpret and evaluate extracted patterns and models, to determine the patterns that can be considered as new knowledge, and to draw conclusions. Still, it should be noted that while evaluation is often mainly technically oriented, the interpretation of results requires close collaboration with domain experts.

Good examples of the knowledge discovery systems that follow Fayyad's view on DM as the process are: SPSS Clementine (Clementine User Guide, Version 5, 1998), SGI Mineset (Brunk *et al.*, 1997), and IBM Intelligent Miner (Tkach, 1998).

2.1.2 Critical issues and perspectives in KDD systems

Numerous KDD systems have recently been developed. At the beginning of this millennium there exist about 200 tools that could perform several tasks (such as clustering, classification, and visualization) for specialized applications ("vertical solutions") (Piatetsky-Shapiro, 2000). This growing trend towards vertical solutions in DM (Fayyad & Uthurusamy, 2002) has been associated with the third generation of DM systems.

The next-generation database mining systems are aimed to manage KDD applications just the way SQL-based systems successfully manage business applications. These systems should integrate the data mining and database subsystems and automate (as far as needed) all the steps of the whole KDD process. These systems should be able to discover knowledge by combining several available KDD techniques. An essential part of the integrated KDD-process is the subpart that enables situation-dependent selection of appropriate KDD technique(s) at every step of a KDD process.

Because of the increasing number of such "vertical solutions" and the possibility to accumulate knowledge from these solutions, there is a growing potential for appearance of next-generation database mining systems to manage KDD applications. While today's algorithms tend to be fully automatic and therefore fail to allow guidance from knowledgeable users at key stages in the search for data regularities, the researchers and the developers, who are involved in the creation of the next generation data mining tools, are motivated to provide a broader range of automated steps in the data mining process and make this process more mixed-initiative. In this process human experts collaborate more closely with the computer to form hypotheses and test them against the data. Moreover, nowadays some initiatives to standardize definition of data mining techniques and the process of knowledge discovery, to provide API are gaining in strength (Grossman *et al.*, 2002). Good examples are: the Predictive Model Markup Language (PMML, 2004) that is an XML-based language which provides a way for applications to define statistical and data mining models and to share models between PMML compliant applications; the SQL Multimedia and Applications Packages Standard (Melton & Eisenberg, 2001), which specifies SQL interface to data mining applications and services, and provides an API for data mining applications to access data from SQL/MM-compliant relational databases; the Java Specification Request-73 (JSR, 2004) that defines a pure Java API supporting the building of data mining models and the creation, storage, and access to data and metadata; the Microsoft-supported OLE DB for DM defining an API for data mining for Microsoft-based applications (OLE DB, 2004); the CRoss-Industry Standard Process for Data Mining (CRISP-DM, 2004) capturing the data mining process from business problems to deployment of the knowledge gained during the process.

2.2 Supervised learning and classifiers

A typical data mining task is to explain and predict the value of some attribute of the data given a collection of fields of some tuples with known attribute values (Chan & Stolfo, 1997). This task is often solved with *inductive learning*, the process of building a model from training data. The resulting model is then used to make predictions on previously unseen examples.

2.2.1 Supervised learning: the taxonomy of concepts

The task of placing an *instance* **x** into one of a finite set of possible categories **c** is called *classification* (Aivazyan, 1989). Often an *instance (*also called an *example* or a *case)* is defined by specifying the value of each feature. This is known as *feature-value* (also called *attribute-value*) notation of a data that represents a problem, and may be written as a row vector using the following notation:

$$\mathbf{x} = [v(x_1), v(x_2),\ldots,v(x_d)], \tag{1}$$

where $v(x_i)$ denotes the value of feature (attribute) x_i, and d is the number of features. Features, which take on a value from an unordered set of possible values, are called *categorical* (also called *nominal*). *Continuous* features are used whenever there is a linear ordering on the values, even if they are not truly continuous. The features used to define an instance are paired, as a rule, with an extra categorical feature that is called *class attribute* (also called *output* *attribute*). The range of all possible values of the features of instances is referred to as the *instance space* (also called *example space*).

Typically, instances with a given classification value are used for supervised learning (building classifiers), and are called *training set* (also called *learning set*, or simply a *dataset*). The classifiers are usually applied to instances with unknown (for a classifier) class value, and called *test* (also called *unseen*) instances, which constitute a *test set*. The classification of test instance $C(\mathbf{x}_{test}) \in$ range($\mathbf{c}$) = $\{c_1, \ldots , c_c\}$, where index c is the number of classes, is the process of predicting the most probable c_i. However, in the test set class values should be present so that an evaluator of the classifier is able to check the correctness of the prediction for each test instance.

A common measure of a classifier's performance is *error rate* that is calculated as the percentage of misclassified test instances (Merz, 1998), and *classification accuracy* (also called *generalization performance*) that is the percentage of correctly classified test instances. More generally, the accuracy of a classifier is the probability of correctly classifying a randomly selected instance (Kohavi, 1995b). Classification accuracy measure is used in this thesis to evaluate the performance of a data mining strategy (for example a coupled combination of a FE technique and a classifier).

Classifiers may vary widely from simple rules to neural networks. However, we are mainly interested here in the instance-based, Naïve Bayes and decision-tree learning techniques. These are briefly described in the following sections. These learning techniques are used in experiments when application of FE for supervised learning is analysed.

2.2.2 Instance-based classifier

An instance-based learning algorithm stores a series of training instances in its memory and uses a distance metric to compare new instances to those stored. Prediction on the new instance is based on the instance(s) closest to it (Aha *et al.*, 1991). The simplest and most well studied instance-based learning algorithm is known as the "nearest neighbor" (NN) classifier.

The classifier stores all instances from the training set (this memorisation is hard to refer to as a training/learning phase) and classifies an unseen instance on the base of a similarity measure. The distance from the unseen instance to all the training instances is calculated and the class label corresponding to the closest training instance is assigned to the example. The most elementary version of the algorithm is limited to continuous features with the Euclidean distance metric. Categorical features are binarised and then treated as numerical.

A more sophisticated version of the nearest neighbor classifier returns the most frequent class among the *k* closest training examples (denoted kNN) (Aha *et al.*, 1991). A weighted average of the nearest neighbors can be used, for example in weighted nearest neighbor (WNN) (Cost & Salzberg, 1993). Given a specific instance that shall be classified, the weight of an example increases with increasing similarity to the example to be classified. In this thesis we use the IBk instance-based learning algorithm from WEKA machine learning library in Java (Witten & Frank, 2000), and the PEBLS instance-based learning algorithm (Cost & Salzberg, 1993), and WNN classifier implemented within MLC++ – machine learning library in C++ (Kohavi *et al.*, 1996) for dynamic integration of classifier (see Section 2.3).

A major problem of the simple approach of kNN is that the vector distance will not necessarily be the best measure for finding intuitively similar examples, especially if irrelevant attributes are present.

2.2.3 Naïve-Bayes classifier

The Naïve-Bayes (NB) classifier (John, 1997) uses Bayes rule to predict the class of a previously unseen example, given a training set. Bayes' theorem defines how to compute the probability of each class given the instance, assuming the features are conditionally independent given the class. The chosen class is the one that maximizes the conditional probability:

$$P(c_i \mid \mathbf{x}_{test}) = \frac{P(c_i)}{P(\mathbf{x}_{test})} \prod_{j=1}^{k} P(x_j \mid c_i) \tag{2}$$

where c_i is the i-th class, $\mathbf{x}_{test}$ is a test example, and P($A \mid B$) is the conditional probability of A given B, P($\mathbf{x}_{test} \mid c_i$) is broken down into the product P($x_k \mid c_i$)… P($x_k \mid c_i$), where x_j is the value of the j-th feature in the example $\mathbf{x}_{test}$.

More sophisticated Bayesian classifiers were developed, for example by John (1997), but only the Naïve-Bayes classifier is used in the experiments in this study.

The Naïve-Bayes classifier relies on an assumption that is rarely valid in practical learning problems, and therefore has traditionally not been the focus of research. It has sometimes been used as the base against which more sophisticated algorithms are compared. However, it has been recently shown that, for classification problems where the predicted value is categorical, the independence assumption is less restrictive than might be expected (Domingos & Pazzani, 1996; Domingos & Pazzani, 1997; Friedman, 1997). Domingos and Pazzani (1997) have presented a derivation of necessary and sufficient conditions for the optimality of the simple Bayesian classifier showing that it can be optimal even when the independence assumption is violated by a wide margin. They showed that although the probability estimates that the Naïve-Bayes classifier produces can be inaccurate, the classifier often assigns maximum probability to the correct class.

2.2.4 C4.5 Decision Tree classifier

Decision tree learning is one of the most widely used inductive learning methods (Breiman *et al.*, 1984; Quinlan, 1996). A *decision tree* is represented as a set of nodes and arcs. Each node usually 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.

A decision tree is usually induced using “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.

The requirement that all the data be correctly classified may result in an overly complex decision tree. Extra nodes may be added in response to minor variations in the data. The problem of being overly sensitive to minor fluctuations in the training data is known as *overfitting*, and it is a general problem for all learning algorithms. A common strategy for avoiding overfitting in decision trees is to “prune” away those subtrees of the decision tree, which improves generalization performance on a too small set of pruning validation examples.

The decision tree learning algorithm used in this thesis is WEKA’s implementation of the C4.5 decision tree learning algorithm (Quinlan, 1993), which is the most widely used decision tree learning approach. C4.5 uses gain ratio, a variant of mutual information, as the feature selection measure. C4.5 prunes by using the upper bound of a confidence interval on the resubstitution error as the error estimate; since nodes with fewer instances have a wider confidence interval, they are removed if the difference in error between them and their parents is not significant (Quinlan, 1993).

2.3 Dynamic integration of classifiers

Recently the integration of classifiers (or ensemble of classifiers) has been under active research in machine learning (Dietterich, 1997), and different ensemble approaches have been considered (Chan & Stolfo, 1997). The integration of base classifiers into ensemble has been shown to yield higher accuracy than the most accurate base classifier alone in different real-world problems (Merz, 1998).

In general the process of ensemble of classifiers construction can be considered in the following way (see Figure 2). A set of base classifiers is formed during the learning phase. Each base classifier in the ensemble is trained using training instances of the corresponding training subset. During the integration phase an integration model of classifiers that allows combining the results produced by a set of selected base classifiers is constructed. The integration model produces the final classification of the ensemble.

Use of an ensemble of classifiers gives rise to two basic questions: (1) what is the set of classifiers (often called base classifiers) that should be generated?; and (2) how should the classifiers be integrated? (Merz, 1998). In this thesis we will be interested in applying FE to improve the integration of classifiers.

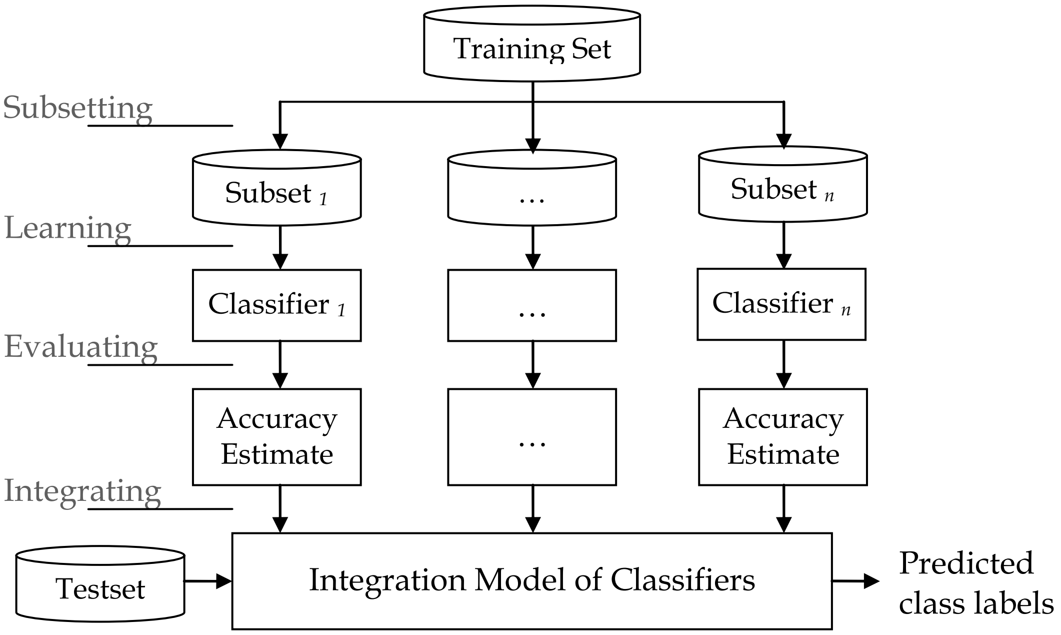


FIGURE 2 The integration of classifiers

2.3.1 Generation of base classifiers

One way of generating a diverse set of classifiers is to use learning algorithms with heterogeneous representations and search biases (Merz, 1998), such as decision trees, neural networks, instance-based learning, etc.

Another approach is to use models with homogeneous representations that differ in their method of search or in the data on which they are trained. This approach includes several techniques for generating base models, such as learning base models from different subsets of the training data. For example, two well-known ensemble methods of this type are bagging and boosting (Quinlan, 1996).

One particular way for building models with homogeneous representations, which proved to be effective, is the use of different subsets of features for each model. For example, in Oza and Tumer (1999) base classifiers are built on different feature subsets, where each feature subset includes features relevant for distinguishing one class label from the others (the number of base classifiers is equal to the number of classes). Finding a set of feature subsets for constructing an ensemble of accurate and diverse base models is also known as ensemble feature selection (Opitz & Maclin, 1999).

Ho (1998) has shown that simple random selection of feature subsets may be an effective technique for ensemble feature selection. This technique is called the RSM and is derived from the theory of stochastic discrimination (Kleinberg, 2000). In the RSM, to construct each base classifier, one randomly selects a subset of features. The RSM has much in common with bagging, but instead of sampling instances, features are sampled (Skurichina & Duin, 2001).

In this thesis we use RSM in ensemble feature selection (see Article VII).

2.3.2 Integration of base classifiers

The challenging problem of integration is to decide what the base classifiers should be or how to combine the results produced by the base classifiers. Two basic approaches have been suggested as a solution to the integration problem: (1) a *combination approach,* where the base classifiers produce their classifications and the final classification is composed using them (Merz, 1998); and (2) a *selection approach,* where one of the base classifiers is selected and the final classification is the result produced by it (Schaffer, 1993).

Techniques for combining or selecting classifiers can be divided into two subsets: *static* and *dynamic*. A static model does not depend on local information. The techniques belonging to the static selection approach propose one "best" method for the whole data space. Usually, better results can be achieved if the classifier integration is done dynamically taking into account the characteristics of each new instance. The basic idea of dynamic integration is that the information about a model's errors in the instance space can be used for learning just as the original instances were used for learning the model. Both theoretical background and practical aspects of dynamic integration can be found in Tsymbal (2002). Gama (1999) showed that the distribution of the error rate over the instance space is not homogeneous for many types of classifiers. Depending on the classifier, the error rate will be more concentrated on certain regions of the instance space than in others.

2.3.3 Dynamic integration approaches used in the study

In this thesis, we will be interested in a dynamic integration approach that estimates the local accuracy of the base classifiers by analyzing their accuracy on nearby instances to the instance to be classified (Puuronen *et al.*, 1999). Instead of directly applying selection or combination as an integration method, cross validation is used to collect information about the classification accuracies of the base classifiers, and this information is then used to estimate the local classification accuracies for each new instance. These estimates are based on the weighted nearest neighbor classification (WNN) (Cost & Salzberg, 1993).

In the study we use three different approaches based on the local accuracy estimates: Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS) (Tsymbal *et al.*, 2001). All these are based on the same local accuracy estimates obtained using WNN. In DS a classifier with the least predicted local classification error is selected. In DV, each base classifier receives a weight that is proportional to the estimated local accuracy of the base classifier, and the final classification is produced by combining the votes of each classifier with their weights. In DVS, the base classifiers with the highest local classification errors are discarded (the classifiers with errors that fall into the upper half of the error interval of the base classifiers) and locally weighted voting (DV) is applied to the remaining base classifiers.

2.4 The curse of dimensionality and dimensionality reduction

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” (Bellman, 1961) pertinent to many learning algorithms, denotes the drastic increase in computational complexity and classification error with data having a large number of dimensions. In this section we consider some interesting properties of high dimensional spaces which motivate the reduction of space dimensionality that could probably be performed without a significant loss of important information for classification. Afterwards we give a brief general categorization of the dimensionality reduction and FE techniques.

2.4.2 Inferences of geometrical, statistical, and asymptotical properties of high dimensional spaces for supervised classification

In this section some unusual or unexpected hyperspace characteristics are discussed in order to show that a higher dimensional space is quite different from a lower dimensional one.

It was shown by Jimenez and Landgrebe (1995) that as dimensionality of a hyperspace increases:

- *The volume of a hypercube concentrates in the corners.*
- *The volume of a hypersphere and a hyperellipsoid concentrate in the outside shell.*

The above characteristics have two important implications for high dimensional data. The first one is that high dimensional space is mostly empty, which implies that multivariate data has usually an intristic lower dimensional structure. As a consequence high dimensional data can be projected to a lower dimensional subspace without loosing significant information, especially in terms of separability among the different classes. The second inference is that normally distributed data will have a tendency to concentrate in the tails, corners, outside shells etc, but not in the “main space”. Support for this tendency can be found in the statistical behaviour of normally and uniformly distributed multivariate data at high dimension.

- *The required number of labelled samples for supervised classification increases as a function of dimensionality.*

It was proved by Fukunaga (1990) that the required number of training samples depends linearly on the dimensionality for a linear classifier and is related to the square of the dimensionality for a quadratic classifier. This fact is very relevant, especially because there exist circumstances where second order statistics are more appropriate than the first order statistics in discriminating among classes in high dimensional data (Fukunaga, 1990). In terms of nonparametric classifiers the situation is even more severe. It has been estimated that as the number of dimensions increases, the sample size needs to increase exponentially in order to have an effective estimate of multivariate densities (Jimenez & Landgrebe, 1995).

It seems, therefore, that original, high dimensional data should contain more discriminative information (comparing to a lower dimensional projection of original data). But at the same time the above characteristics tell us that it is difficult with the current techniques, which are usually based on computations at full dimensionality, to extract such information unless the amount of available labelled data is substantial. The so-called Hughes phenomenon is a concrete example of this – with a limited number of training samples there is a penalty in classification accuracy as the number of features increases beyond some point (Hughes, 1968).

– *For most high dimensional data sets, low linear projections have the tendency to be normal, or a combination of normal distributions, as the dimension increases.*

This is a significant characteristic of high dimensional data that is quite important to the analysis of such data. It has been proved by Hall and Li (1993) that as the dimensionality tends to infinity, lower-dimensional linear projections will approach a normality model. Normality in this case means a normal distribution or a combination of normal distributions.

As a consequence of the introduced properties it is possible to reduce the dimensionality without losing significant information and separability. As the dimensionality increases the increased number of labelled samples is required for supervised classification (where the computation is done at full dimensionality). So, there is a challenge to reduce dimensionality while trying to preserve the discriminative information. Therefore, there is an increasing tendency towards new methods that, instead of doing the computation at full dimensionality, use a lower dimensional subspace(s). This, beside computational benefits, will make the assumption of normality better grounded in reality, yielding a better estimation of parameters, and better classification accuracy.

2.4.3 Dimensionality reduction techniques

In this section we follow the categorization of dimensionality reduction techniques according to the book by Liu (1998).

There are many techniques to achieve dimensionality reduction for data, including multidimensional heterogeneous data, presented by a large number of features of different types. Usually these techniques are divided mainly into dimensionality reduction for optimal data representation and dimensionality reduction for classification, according to their aim. According to the adopted strategy these techniques can also be divided into feature selection and feature transformation (also called feature discovery). The variants of the last one are FE and feature construction. The key difference between feature selection and feature transformation is that during the first process only a subset of original features is selected while the second approach is based on the generation of completely new features. Concerning the distinction between transformation techniques, feature construction implies discovering missing information about the relationships among features by inferring or creating additional features while FE discovers new feature space through a functional mapping.

If a subset of irrelevant and/or redundant features can be recognized and eliminated from the data then feature selection techniques may work well. Unfortunately this is not always easy and sometimes not even possible. This happens because of the fact that a feature subset may be useful in one part of the instance space, and at the same time useless or even misleading in another part of it. And, all methods that just assign weights to the individual features have an essential drawback in that they are insensitive to interacting or correlated features. That is why the transformation of the given representation before weighting the features is often preferable.

In this thesis we are interested in the study of several data mining strategies that apply feature extraction for supervised learning (i.e. for subsequent classification). The next chapter gives a brief introduction to FE techniques used throughout the study.

2.4.4 Feature extraction

Feature extraction (FE) is a process that extracts a subset of new features from the original set by means of some functional mapping. There are several interesting approaches for FE introduced in the literature. Among them are the discriminant analysis, fractal encoding, use of wavelet transformation, use of mutual information, and different types of neural networks (Diamantras & Kung, 1996). The most common technique is still probably the principal component analysis (PCA) and its different variations and extensions (Liu, 1998). We would like to point out that some practitioners from the Pattern Recognition field use the term 'feature extraction' to refer to the process of extracting features from data (Duda *et al.*, 2001).

Besides computational complexity reduction, dimensionality reduction by FE helps also to solve the problem of overfitting, a tendency of a classifier to assign importance to random variations in the data by declaring them important patterns, i.e. the classifier is turned to the contingent, rather than just the constitutive characteristics of the training data (Duda *et al.*, 2001).

In general the FE process requires some domain knowledge and intuition about the problem for the following reasons: different problem areas may require different approaches and domain knowledge also allows restricting the search space and thus helps to effectively find out relevant features (Fayyad, 1996).

However, it should be noticed that the transformed features are often not meaningful in terms of the original domain. Thus, additional constraints on the transformation process are required to guarantee comprehensibility if examination of the transformed classifier is necessary. Sometimes, domain knowledge helps to overcome the problem of interpretability too (Liu, 1998).

According to the availability of the supervised or unsupervised data, FE methods can or cannot use class information. Certainly, this question is crucial for the classification purposes.

Also, one of the key issues in FE is the decision whether to proceed globally over the entire instance space or locally in different parts of the instance space. It can be seen that despite being globally high-dimensional and sparse, data distributions in some domain areas are locally low-dimensional and dense, for example in physical movement systems (Vijayakumar & Schaal, 1997).

2.5 Feature extraction for supervised learning

Generally, FE for supervised learning can be seen as a search process among all possible transformations of the original feature set for the best one, which preserves class separability as much as possible in the space with the lowest possible dimensionality (Aladjem, 1994). In other words we are interested in finding a projection w:

$$\mathbf{y} = \mathbf{w}^T \mathbf{x} \tag{3}$$

where $\mathbf{y}$ is a $k \times 1$ transformed data point (presented using k features), $\mathbf{w}$ is a $d \times k$ transformation matrix, and $\mathbf{x}$ is a $d \times 1$ original data point (presented using d features).

2.5.1 Principal component analysis

Principal Component Analysis (PCA) is a classical statistical method, which extracts a lower dimensional space by analyzing the covariance structure of multivariate statistical observations (Jolliffe, 1986).

The main idea behind PCA is to determine the features that explain as much of the total variation in the data as possible with as few of these features as possible. We are interested in PCA primarily as a widely used dimensionality reduction technique, although PCA is also used for example for the identification of the underlying variables, for visualization of multidimensional data, identification of groups of objects or outliers and for some other purposes (Jolliffe, 1986).

The computation of the PCA transformation matrix is based on the eigenvalue decomposition of the covariance matrix $\mathbf{S}$ (and therefore it is computationally rather expensive).

$$\mathbf{w} \leftarrow eig\_decomposition\left(\mathbf{S} = \sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{m})(\mathbf{x}_i - \mathbf{m})^T \right) \tag{4}$$

where n is the number of instances, $\mathbf{x}_i$ is the i-th instance, and $\mathbf{m}$ is the mean vector of the input data.

Computation of the principal components can be presented with the following algorithm:

1. Calculate the covariance matrix $\mathbf{S}$ from the input data.

2. Compute the eigenvalues and eigenvectors of **S** and sort them in a descending order with respect to the eigenvalues.
3. Form the actual transition matrix by taking the predefined number of components (eigenvectors).
4. Finally, multiply the original feature space with the obtained transition matrix, which yields a lower- dimensional representation.

The necessary cumulative percentage of variance explained by the principal axes is used commonly as a threshold, which defines the number of components to be chosen.

In the case of high-dimensional data PCA is computationally expensive, especially if only a few of the first components are needed. Also, when new data points are observed and the PCA-based model is being updated, the covariance matrix and its eigenvalues require complete recalculation. Therefore, many algorithms for PCA, which extract only the desired number of principal components and which can adapt to new data have been introduced and examined (Weingessel & Hornik, 1998).

2.5.2 The random projection approach

In many application areas like market basket analysis, text mining, image processing etc., dimensionality of data is so high that commonly used dimensionality reduction techniques like PCA are almost inapplicable because of extremely high computational time/cost.

Recent theoretical and experimental results on the use of random projection (RP) as a dimensionality reduction technique have attracted the DM community (Bingham & Mannila, 2001). In RP a lower-dimensional projection is produced by means of transformation like in PCA but the transformation matrix is generated randomly (although often with certain constrains).

The theory behind RP is based on the Johnson and Lindenstrauss Theorem (see for example Dasgupta & Gupta, 2003) that says that any set of n points in a d-dimensional Euclidean space can be embedded into a k-dimensional Euclidean space – where k is logarithmic in n and independent of d – so that all pairwise distances are maintained within an arbitrarily small factor (Achlioptas, 2001). The basic idea is that the transformation matrix has to be orthogonal in order to protect data from significant distortions and try to preserve distances between the data points. Generally, orthogonalization of the transformation matrix is computationally expensive, however, Achlioptas (2001) showed a very easy way of defining (and also implementing and computing) the transformation matrix for RP. So, according to Achlioptas (2001) the transformation matrix **w** can be computed simply either as:

$$w_{ij} = \sqrt{3} \cdot \begin{cases} +1 & \text{with probability } 1/6 \\ 0 & \text{with probability } 2/3 \\ -1 & \text{with probability } 1/6 \end{cases}, \quad \text{or} \quad w_{ij} = \begin{cases} +1 & \text{with probability } 1/2 \\ -1 & \text{with probability } 1/2 \end{cases} \qquad (5)$$

RP as a dimensionality reduction technique was experimentally analyzed on image (noisy and noiseless) and text data (a newsgroup corpus) by Bingham and Mannila (2001). Their results demonstrate that RP preserves the similarity of data vectors rather well (even when data is projected onto relatively small numbers of dimensions).

Fradkin and Madigan (2003) performed experiments on 5 different data sets with RP and PCA for inductive supervised learning. Their results show that although PCA predictively outperformed RP, RP is a rather useful approach because of its computational advantages. The authors also indicated a trend in their results, namely that the predictive performance of RP is improved with increased dimensionality when combining with the right learning algorithm. It was found that for those 5 data sets RP is suited better for nearest neighbour methods, where preserving distance between data points is more important than preserving the informativeness of individual features, in contrast to the decision tree approaches where the importance of these factors is reversed. However, further experimentation was encouraged.

Related work on RP includes use of RP as preprocessing of textual data, for further LSI (Papadimitriou *et al.*, 1998), for indexing of audio documents with further LSI and use of SOM (Kurimo, 1999), for nearest-neighbor search in a high dimensional Euclidean space (Kleinberg, 1997; Indyk & Motwani, 1998), for learning high-dimensional Gaussian mixture models (Dasgupta 1999; 2000).

In general, the use of random methods (with regard to manipulations on features space) has a strong and lengthy tradition in DM community mainly because of practical success of random forests (Breiman, 2001), and the random subspace method (RSM) (Ho, 1998).

2.5.3 Class-conditional feature extraction

Although PCA is still probably the most popular FE technique, it has a serious drawback, i.e., giving high weights to features with higher variabilities, irrespective of whether they are useful for classification or not. This may give rise to a situation where the chosen principal component corresponds to an attribute with the highest variability but has no discriminating power (Oza & Tumer, 1999).

A usual approach to overcome the above problem is to use some class separability criterion (Aladjem, 1994), for example the criteria defined in Fisher's linear discriminant analysis (Fisher, 1936) and based on the family of functions of scatter matrices:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}}, \tag{6}$$

where $\mathbf{S}_B$ in the parametric case is the between-class covariance matrix that shows the scatter of the expected vectors around the mixture mean , and $\mathbf{S}_W$ is the within-class covariance, that shows the scatter of samples around their respective class expected vectors. Thus,

$$\mathbf{S}_W = \sum_{i=1}^{c} n_i \sum_{j=1}^{n_i} (\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})(\mathbf{x}_j^{(i)} - \mathbf{m}^{(i)})^T , \quad \text{and} \quad \mathbf{S}_B = \sum_{i=1}^{c} n_i (\mathbf{m}^{(i)} - \mathbf{m})(\mathbf{m}^{(i)} - \mathbf{m})^T , \qquad (7)$$

where c is the number of classes, n_i is the number of instances in a class i, $\mathbf{x}_j^{(i)}$ is the j-th instance of i-th class, $\mathbf{m}^{(i)}$ is the mean vector of the instances of i-th class, and $\mathbf{m}$ is the mean vector of all the input data.

The total covariance matrix shows the scatter of all samples around the mixture mean. It can be shown analytically that this matrix is equal to the sum of the within-class and between-class covariance matrices (Fukunaga, 1990). In this approach the objective is to maximize the distance between the means of the classes while minimizing the variance within each class. A number of other criteria were proposed by Fukunaga (1990).

The criterion (6) is optimized using the simultaneous diagonalization algorithm (see for example Fukunaga, 1990). The basic steps of the algorithm include eigenvalues decomposition of $\mathbf{S}_W$; transformation of original space to intermediate $\mathbf{x}_W$ (whitining); calculation of $\mathbf{S}_B$ in $\mathbf{x}_W$; eigenvalues decomposition of $\mathbf{S}_B$ and then transformation matrix $\mathbf{w}$ finally can be produced by a simple multiplication:

$$\begin{aligned} &\mathbf{w}_{S_W} \leftarrow eig\_decomposition(S_W), \quad x_W = \mathbf{w}_{S_W} x; \\ &\mathbf{w}_{S_B} \leftarrow eig\_decomposition(S_B \mid x_W) \\ &\mathbf{w} = \mathbf{w}_{S_W} \mathbf{w}_{S_B} \end{aligned} \qquad (8)$$

The parametric approach considers one mean for each class and one total mixture mean when computing the between class covariance matrix. Therefore, there is a fundamental problem with the parametric nature of the covariance matrices. The rank of $\mathbf{S}_B$ is at most the *number of classes-1*, and hence no more than this number of new features can be obtained through the FE process.

The nonparametric method overcomes this problem by trying to increase the number of degrees of freedom in the between-class covariance matrix, measuring the between-class covariances on a local basis. The k-nearest neighbor (kNN) technique is used for this purpose. In the nonparametric case the between-class covariance matrix is calculated as the scatter of the samples around the expected vectors of other classes′ instances in the neighborhood:

$$\mathbf{S}_B = \sum_{i=1}^{c} n_i \sum_{k=1}^{n_i} w_{ik} \sum_{\substack{j=1 \\ j \neq i}}^{c} (\mathbf{x}_k^{(i)} - \mathbf{m}_{ik*}^{(j)})(\mathbf{x}_k^{(i)} - \mathbf{m}_{ik*}^{(j)})^T \qquad (9)$$

where $\mathbf{m}_{ik*}^{(j)}$ is the mean vector of the n*NN* instances of the j-th class, which are nearest neighbors to $\mathbf{x}_k^{(i)}$. The coefficient w_{ik} is a special weighting coefficient, which shows the importance of each summand in (9). The goal of this coefficient is to assign more weight to those elements of the matrix which involve instances lying near the class boundaries and are thus more important for classification.

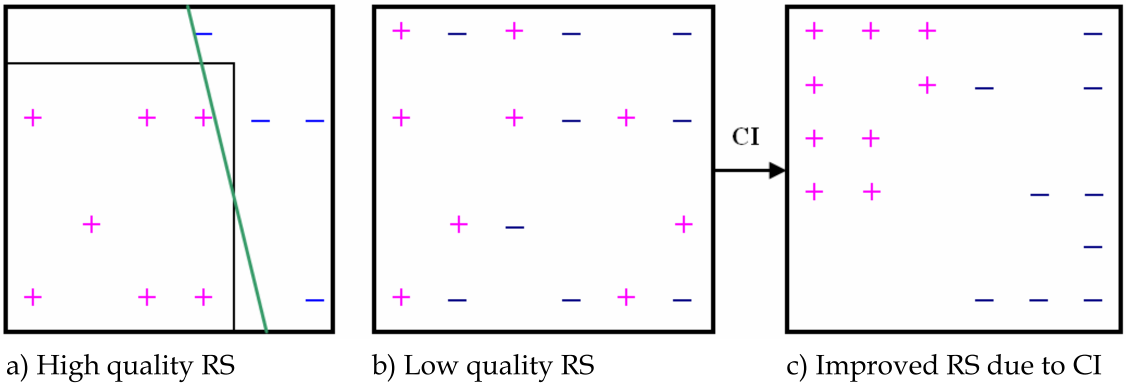
Thus, a nonparametric approach is potentially more efficient than a parametric one, because it constructs flexible bounds between classes. However, the computational complexity of a nonparametric approach is higher. For further details on these class-conditional approaches, please see Pechenizkiy *et al.*, (2004).

2.5.4 FE for supervised learning as an analogy to 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 other one is concerned with generating hypothesis in the found space (Michalski, 1997).

In Figure 3 we can see two two-class ("+" and "–") problems – with a) high quality and b) low quality representation spaces (RS). In a) points marked by "+" are easily separated from the points marked by "–" using a 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, whereas constructive induction approach is to search for a better representation space where the groups are much better separated as is the situation in c).

Constructive induction systems view 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. Bloedorn *et al.* (1993) consider meta-rules construction from meta-data to guide the selection of the operators.



a) High quality RS

b) Low quality RS

c) Improved RS due to CI

FIGURE 3 High vs. low quality representation spaces (RS) for concept learning (Arciszewski *et al.*, 1995, 9)

2.6 Selecting representative instances for FE

When a data set contains a huge number of instances, some sampling approach is commonly applied to address the computational complexity of knowledge discovery processes. In this thesis we are interested in the study of sample reduction effect on the considered FE techniques with regard to the classification performance of a supervised learner.

We use four different strategies to select samples: (1) random sampling, (2) stratified random sampling, (3) *kd*-tree based selective sampling, and (4) stratified sampling with *kd*-tree based selection.

2.6.1 Random sampling

Random sampling and stratified random sampling are the most commonly applied strategies as they are straightforward and extremely fast. In random sampling the information about the distribution of the instances by classes is disregarded. So, defining the percentage *p* of the total number of training set *N* instances to take, we select S = 0.01**N***p* sample for building FE model and consequent supervised learning (Figure 4).

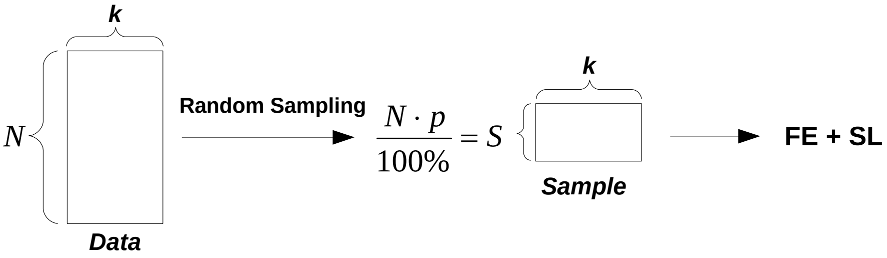


FIGURE 4 Random sampling

Intuitively, stratified sampling, which randomly selects instances from each chunk (group of instances) related to the corresponding class separately, might be preferable if we have the supervised learning process in mind.

2.6.2 Stratified random sampling

Figure 5 presents the basic idea of stratified random sampling. Conceptually, the first step (that certainly can be omitted algorithmically) is to divide data into *c* (equal to number of classes) chunks. Then, random sampling is applied for each data chunk.

However, the assumption that instances are not uniformly distributed and some instances are more representative than others (Aha *et al.*, 1991) motivates to apply a selective sampling approach. Thus, the main idea of selective sampling is to identify and select representative instances, so that fewer instances are needed to achieve similar (or even better) performance. The common approach to selective sampling is data partitioning (or data indexing) that is aimed to find some structure in data and then to select instances from each partition of the structure. Although there exist many data partitioning techniques (see for example (Gaede & Gunther, 1998) for an overview), we choose *kd*-tree for our study because of its simplicity, and wide use.

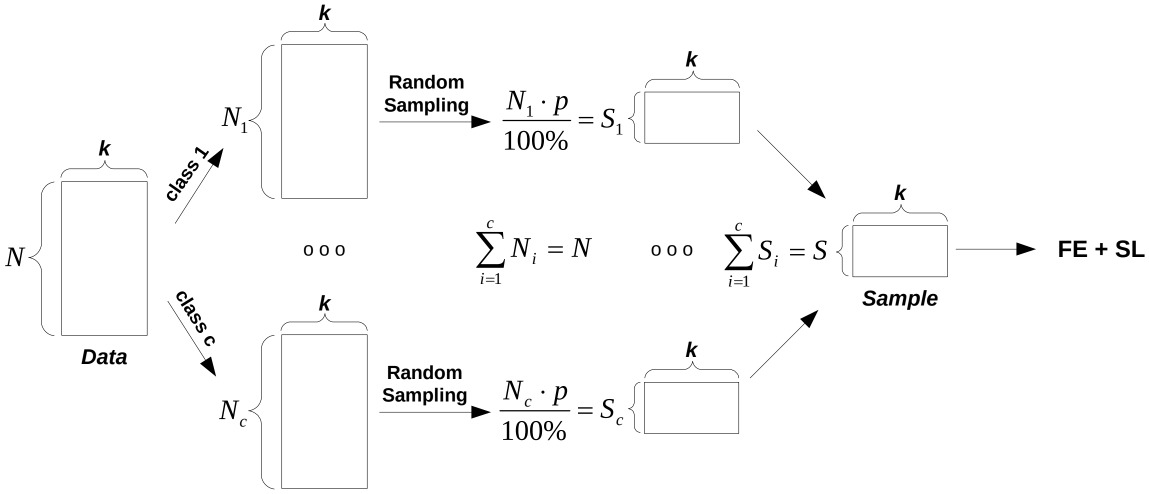


FIGURE 5 Stratified random sampling

2.6.3 *kd*-Tree based sampling

A *kd*-tree is a generalization of the simple binary tree which uses *k* features instead of a single feature to split instances in a multi-dimensional space (Gaede & Gunther, 1998). The splitting is done recursively in each of the successor nodes until the node contains no more than a predefined number of instances (called bucket size) or cannot be split further. The order in which features are chosen to split can result in different *kd*-trees. As the goal of partitioning for selective sampling is to split instances into different (dissimilar) groups, a splitting feature is chosen if the data variance is maximized along the dimension associated with the splitting feature.

In Figure 6 the basic idea of selective sampling is presented graphically. First, a *kd*-tree is constructed from data, then a defined percentage of instances is selected from each leaf of the tree and added to the resulting sample to be used for FE models construction and supervised learning.

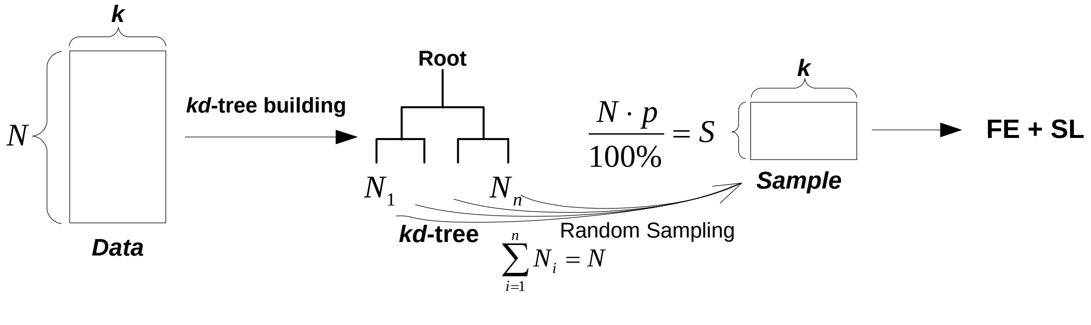


FIGURE 6 *kd*-Tree based selective sampling

2.6.4 Stratified sampling with *kd*-tree based selection of instances

Potentially, the combination of these approaches, so that both class information and information about data distribution are used, might be useful. This idea is presented in Figure 7. It can be seen from the figure that in this approach instead of constructing one global tree, several local *kd*-trees for each data chunk related to certain class are constructed.

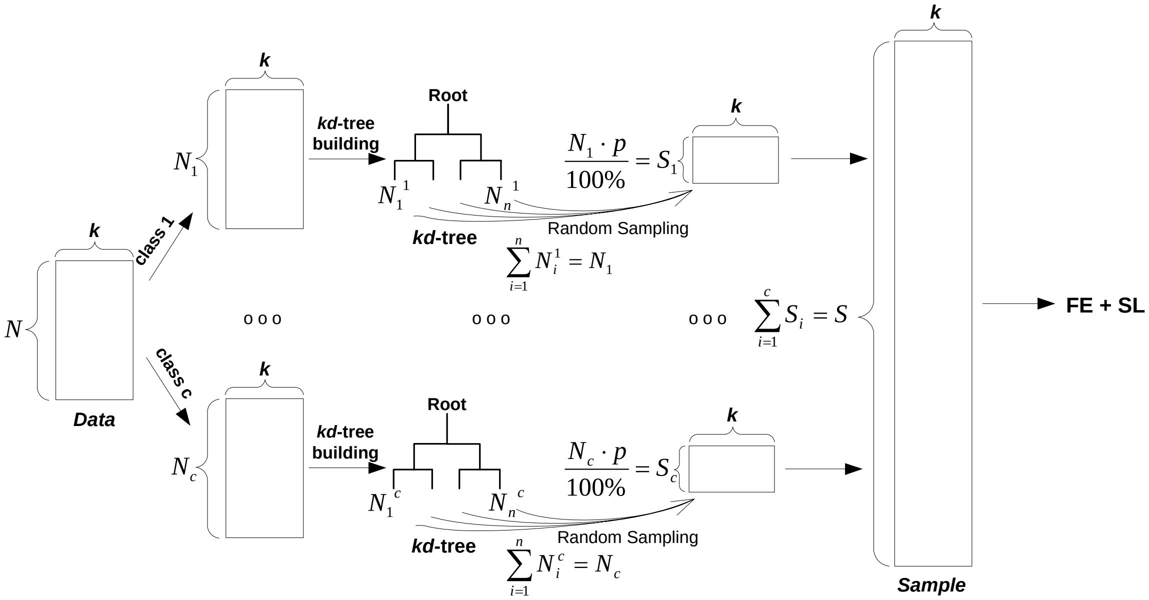


FIGURE 7 Stratified sampling with *kd*-tree based selection of instances

3 RESEARCH PROBLEM

The idea of learning from data is far from being new. However, perhaps due to developments in the Information Technology, Database Management and the huge increase of data volumes being accumulated in databases the interest in DM has become very intense. Numerous DM algorithms have recently been developed to extract knowledge from large databases. Nevertheless, nowadays the complexity of real-world problems, high dimensionality of data being analyzed and poor representation spaces due to presence of many irrelevant or indirectly relevant features challenge learning algorithms. It is commonly accepted that just by pushing a button someone should not expect useful results to appear.

FE is an effective data pre-processing step aimed to reduce the dimensionality and to improve representation space of the problem at consideration. There exists a strong theoretical background on FE (techniques) and supervised learning (SL) from applied statistics, pattern recognition and related field. However, many issues related to the integration of FE and SL processes have not been studied intensively perhaps due to the rare emphasis on DM/KDD as an iterative (and interactive) process (remember Figure 1) and due to the absence of relatively large collection of benchmark data sets to conduct extensive experimental study.

The purpose of this study is to develop theoretical background and practical aspects of FE as means of (1) dimensionality reduction, and (2) representation space improvement for SL in knowledge discovery systems. The focus is on applying conventional Principal Component Analysis (PCA) and two class-conditional approaches considered in Section 2.5 for two targets: (1)

for a base level classifier construction, and (2) for dynamic integration of the base level classifiers.

The main dimensions of research issues related to FE for SL can be recognized from Figure 8. Each instance of a dataset has associated class-value and, originally, features' values $x_1 \ldots x_k$. By means of feature transformation new features can be extracted (constructed) by some functional mapping. Thus each instance will have additional values of features $y_1 \ldots y_m$ if they are added to the original ones. Intuitively, this might be useful when the number of original features is too small. By means of feature selection process a number of original (hopefully, redundant and irrelevant) features can be eliminated. In general, different number of features can be selected from k and m for each data cluster (partition). Besides construction and selection of the most relevant features, the most representative instances for each class can be selected if the sample size is relatively large.

The rest of this Chapter is organized so that each section corresponds to one of the recognized research questions: *"How important is it to use class information in the FE process?"* (Section 3.1); *"Is FE a data- or hypothesis-driven constructive induction?"* (Section 3.2); *"Is FE for dynamic integration of base-level classifiers useful in a similar way as for a single base-level classifier?"* (Section 3.3); *"Which features – original, extracted or both – are useful for SL?"* (Section 3.4); *"How many extracted features are useful for SL?"* (Section 3.5); *"How to cope with the presence of contextual features in data, and data heterogeneity?"* (Section 3.6); *"What is the effect of sample reduction on the performance of FE for SL?"* (Section 3.7); *"When is FE useful for SL?"* (Section 3.8); *"What is the effect of FE on interpretability of results and transparency of SL?"* (Section 3.9); *"How to make a decision about the selection of the appropriate DM strategy (particularly, the selection of FE and SL techniques) for a problem at consideration?"* (Section 3.10).

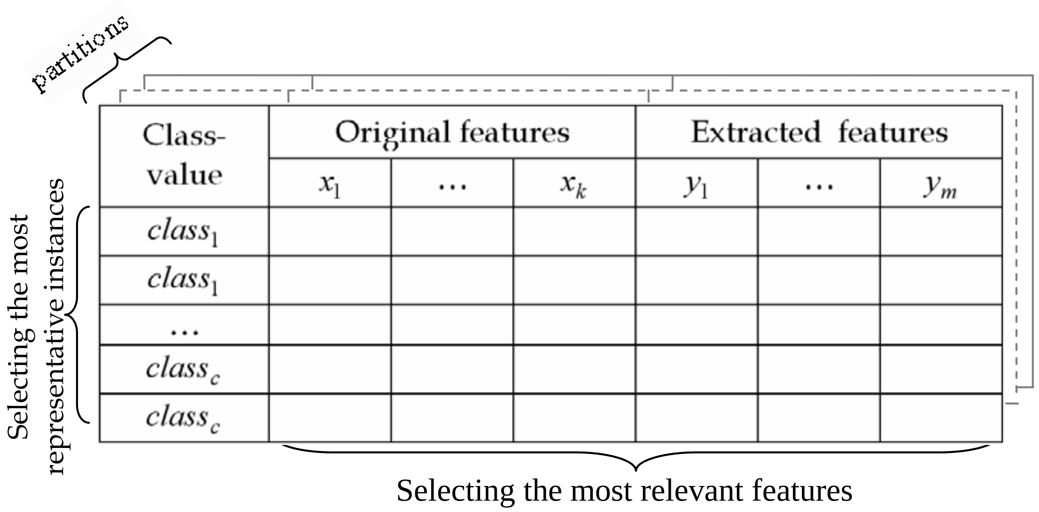


FIGURE 8 The feature-values representations of the instances

3.1 How important is it to use class information in the FE process?

Giving large weights to features with higher variabilities, irrespective of whether they are useful for classification or not can be dangerous. This may give rise to a situation where the chosen principal component corresponds to an attribute with the highest variability but has no discriminating power as shown in Figure 9 (Oza & Tumer, 1999).

Our goal is to study the performance of conventional PCA for SL and compare it with class-conditional parametric and nonparametric approaches presented in Section 2.5.

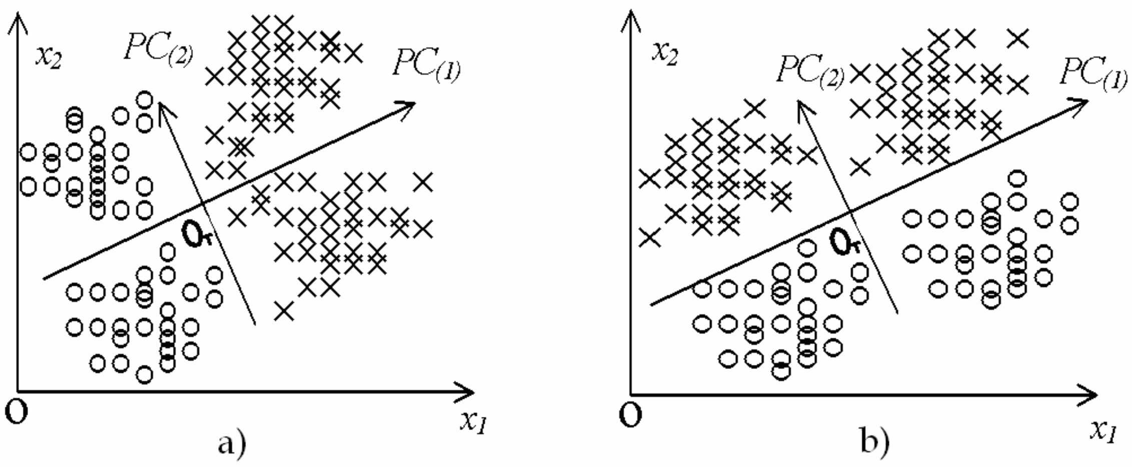


FIGURE 9 PCA for classification: a) effective work of PCA, b) the case where an irrelevant principal component ($PC_{(1)}$) was chosen from the classification point of view (**O** denotes the origin of the initial feature space x_1, x_2 and $\mathbf{O}_T$ – the origin of the transformed feature space $PC_{(1)}$, $PC_{(2)}$)

3.2 Is FE a data- or hypothesis-driven constructive induction?

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 (Arciszewski *et al.*, 1995).

We consider FE for SL as an analogy of constructive induction (classification). Indeed, what we are trying to achieve by means of FE is the most appropriate data representation for the subsequent SL.

One approach is to select and perform FE, keeping in mind the subsequent classification, and then perform the selection of a classifier (Figure 10). However, another approach – the selection of a combination of a FE technique and a classifier may be sufficient. In this case FE and classification cannot be separated into two different independent processes (Figure 11).

Our goal is to address this intuitive reasoning and to study if FE techniques have different effects on the performance of different widely used classifier like Naïve Bayes, C4.5 and kNN.

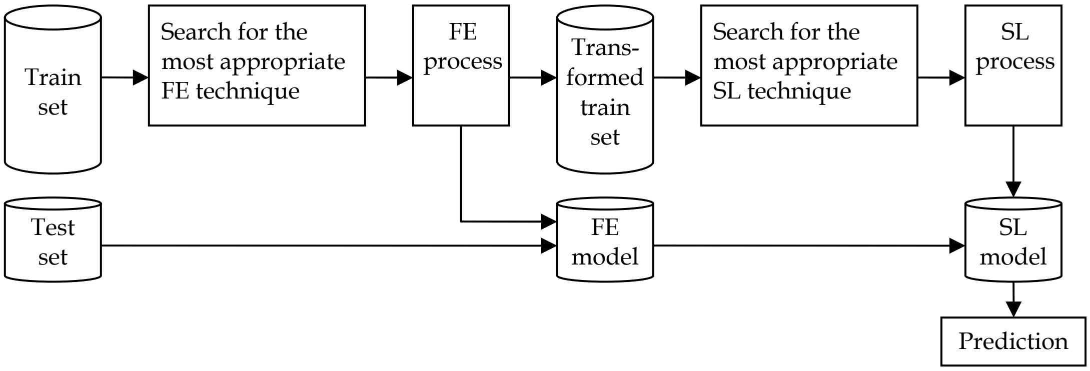

Train set
Search for the most appropriate FE technique
FE process
Trans-formed train set
Search for the most appropriate SL technique
SL process
Test set
FE model
SL model
Prediction

FIGURE 10 Independent searches for the most appropriate FE and SL techniques

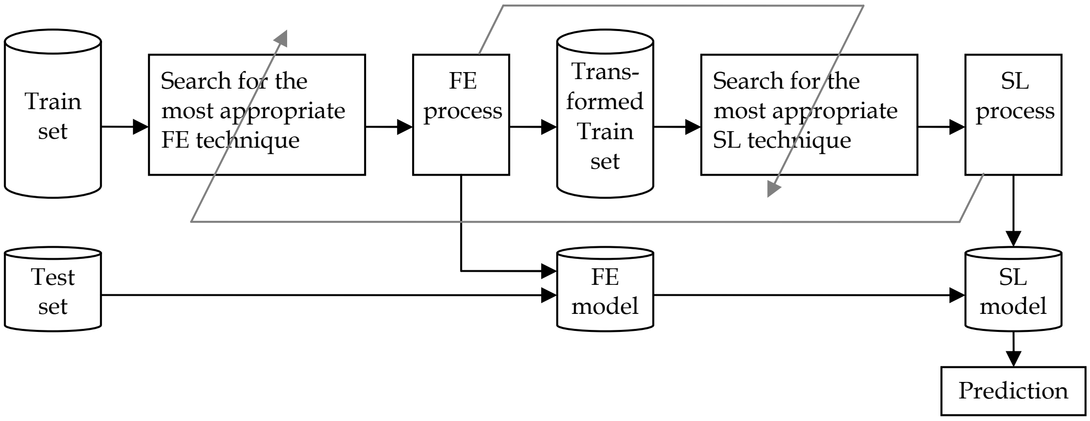

Train set
Search for the most appropriate FE technique
FE process
Trans-formed Train set
Search for the most appropriate SL technique
SL process
Test set
FE model
SL model
Prediction

FIGURE 11 The joint search for a combination of the most appropriate FE and SL techniques

3.3 Is FE for dynamic integration of base-level classifiers useful in a similar way as for a single base-level classifier?

Recent research has shown the integration of multiple classifiers to be one of the most important directions in machine learning and data mining (Dietterich, 1997). Generally, the whole space of original features is used to find the neighborhood of a new instance for local accuracy estimates in dynamic integration. We propose to use FE in order to cope with the curse of dimensionality in the dynamic integration of base-level classifiers (Figure 12).

Our main hypothesis to test is that with data sets where FE improves classification accuracy when employing a single classifier (such as kNN), it would also improve classification accuracy when a dynamic integration approach is employed. Conversely, with data sets where FE decreases (or has no effect to) classification accuracy with the use of a single classifier, then FE will also decrease (or will have no effect to) classification accuracy when employing a dynamic integration approach.

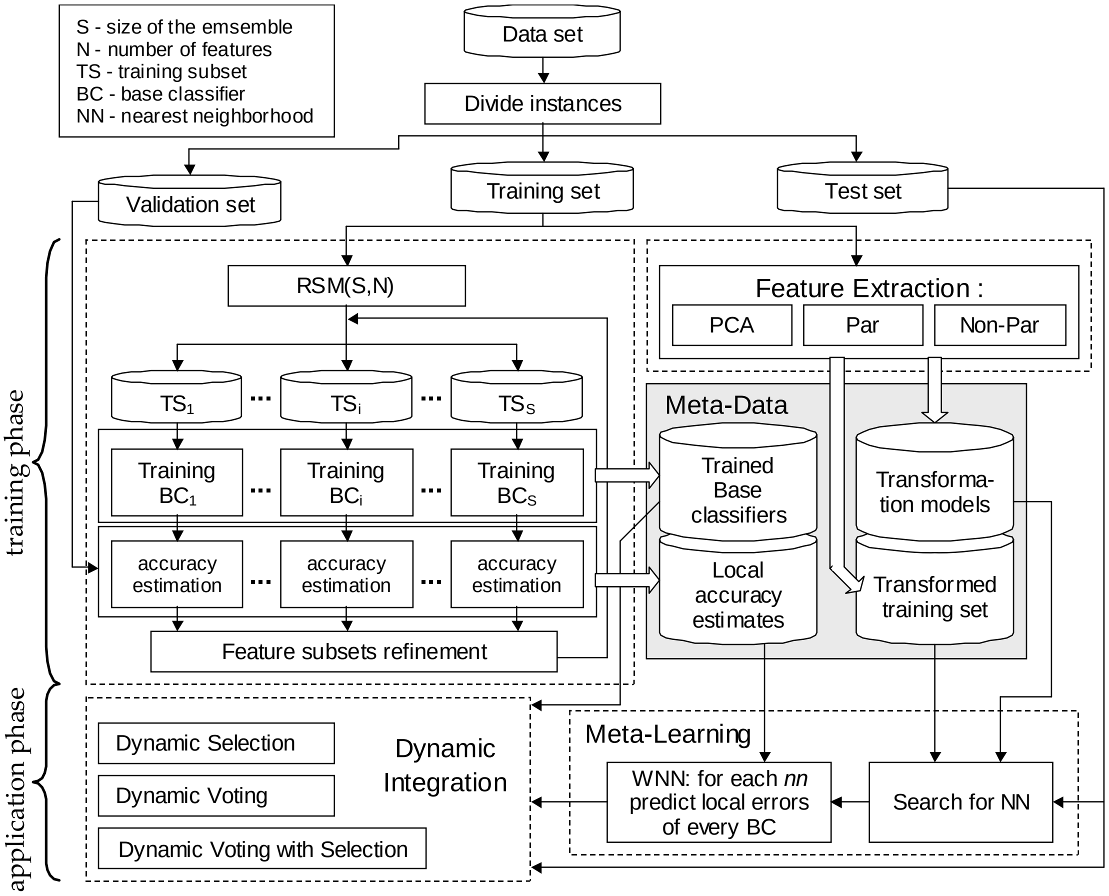


FIGURE 12 Scheme of the FEDIC approach (see Article VII for the detailed description)

3.4 Which features – original, extracted or both – are useful for SL?

FE is often seen as a dimensionality reduction technique. An alternative is to see FE as a useful transformation that leads to representation space improvement, for example due to elimination of correlated and uninformative features and construction of uncorrelated and informative ones instead. However, when the number of original features is relatively small, some new features produced by means of FE process may give additional value for the set of original ones.

Popelinsky (2001) used some transformed features as additional ones for a decision-tree learner, instance-based learner and Naïve Bayes learner, and found that adding principal components to the original dataset results in a decrease in error rate for many datasets for all three learners, although a decrease of error rate was significant only for the instance-based learner. Another interesting result was that for a decision-tree the learner decrease of error rate could be achieved without increasing complexity of the decision tree.

Our goal is to study advantages of using either extracted features or both original and extracted features for SL with regard to the FE approaches considered in Section 2.5.

3.5 How many extracted features are useful for SL?

When transformation matrix **w** for (3) (Section 2.5) is found, someone is interested in deciding how many extracted features to take and what is the best subset of orthogonally transformed features for SL.

One common method is to introduce some threshold, for example variance accounted by a component to be selected. This results in selecting principal components which correspond to the largest eigenvalues. The problem with this approach is that the magnitude of eigenvalue depends on data variance only and has nothing to do with class information. Jollife (1986) presents several real-life examples where principal components corresponding to the smallest eigenvalues are correlated with the output attribute. So, principal components important for classification may be excluded because they have small eigenvalues. In Figure 9 another simple example of such a situation was shown. Nevertheless, criteria for selecting the most useful transformed features are often based on variance accounted by the features to be selected.

An alternative approach is to use a ranking procedure and select principal components that have the highest correlations with the class attribute. Although this makes intuitive sense, there is criticism of such an approach. Almoy (1996) showed that this alternative approach worked slightly worse than using components with the largest eigenvalues in the prediction context.

Our goal is to analyze the performance of different FE techniques when different number of extracted features is selected for SL.

3.6 How to cope with the presence of contextual features in data, and data heterogeneity?

For some datasets a feature subset may be useful in one part of the instance space, and at the same time it may be useless or even misleading in another part of it. Therefore, it may be difficult or even impossible for some problem domains to remove irrelevant and/or redundant features from a data set and leave only useful ones by means of global FS. However, if it is possible to find local homogeneous regions of heterogeneous data, there are more chances to successfully apply FS. For FE the decision whether to proceed globally over the entire instance space or locally in different parts of the instance space is also one of the key issues. It can be seen that despite being globally high-dimensional and sparse, data distributions in some domain areas are locally low-dimensional and dense, for example in physical movement systems.

One possible approach for local FS or local FE would be clustering (partitioning) of the whole dataset into smaller regions. Generally, different clustering techniques can be used for this purpose, for example the k-means or EM techniques (Duda *et al.*, 2001). However, in this thesis our focus is on a possibility to apply so-called *natural clustering* aimed to use contextual features for splitting whole heterogeneous data into more homogeneous clusters. Often such (possibly hierarchical) contextual features may be constructed by domain experts. Usually, contextual (or environmental) features are assumed to be features that are not useful for classification by themselves but are useful in combination with other (context-sensitive) features (Turney, 1996).

Our goal is to analyse the performance of FE and SL when applied globally to the whole data and locally within natural clusters on a data set which is likely heterogeneous and contains contextual features.

3.7 What is the effect of sample reduction on the performance of FE for SL?

When a data set contains a huge number of instances, some sampling strategy is commonly applied before the FE or SL processes to reduce their computational time and cost.

In this thesis we are interested to study the effect of sample reduction on FE for SL. The goal is to study if it is important to take into account class information (to apply some sort of stratification) and preserve variance in data or select the most representative instances during the sampling process. The intuitive hypothesis is that the type of sampling approach is not important when the selected sample size is relatively large. However, it might be important to take into account both class information and information about data distribution when the sample size to be selected is small.

Another goal is to find out if sample reduction has different effect on different FE approaches.

3.8 When is FE useful for SL?

An important issue is how to decide (for example analyzing the space of original features or meta-data if available) whether a PCA-based FE approach is appropriate for a certain problem or not. Since the main goal of PCA is to extract new uncorrelated features, it is logical to introduce some correlation-based criterion with a possibility to define a threshold value. One of such criteria is the Kaiser-Meyer-Olkin (KMO) criterion that accounts for both total and partial correlation:

$$KMO = \frac{\sum_i \sum_j r_{ij}^2}{\sum_i \sum_j r_{ij}^2 + \sum_i \sum_j a_{ij}^2}, \tag{10}$$

where $r_{ij} = r(x^{(i)}, x^{(j)})$ is the element of the correlation matrix $\mathbf{R}$ and a_{ij} are the elements of **A** (partial correlation matrix), and

$$a_{ij.X^{(i,j)}} = \frac{-R_{ij}}{\sqrt{R_{ii}R_{jj}}}, \tag{11}$$

where $a_{ij.X^{(i,j)}}$ is a partial correlation coefficient for $x^{(i)}$ and $x^{(j)}$, when the effect of all the other but *i* and *j* features denoted as $X^{(i,j)}$ is fixed (controlled), and R_{kl} is an algebraic complement for r_{kl} in the determinant of the correlation matrix $\mathbf{R}$.

It can be seen that if two features share a common factor with other features, their partial correlation a_{ij} will be small, indicating the unique variance they share. And then, if a_{ij} are close to zero (the features are measuring a common factor) KMO will be close to one, while if a_{ij} are close to one (the variables are not measuring a common factor) KMO will be close to zero.

Generally, it is recommended to apply PCA for a data set only if KMO is greater than 0.5. Popelinsky (2001) recommended PCA for meta-learning tasks if KMO is greater than 0.6.

Our goal is to analyze the appropriateness of criterion (10) for the decision-making process on usefulness of FE for a problem under consideration.

3.9 Interpretability of the extracted features

Interpretability context in the DM applications commonly refers to the issue whether a classifier is easy to understand. It is commonly accepted that rule-based classifiers like a decision tree and associative rules are very easy to interpret, and neural networks and other connectionist and "black-box" classifiers have low interpretability. kNN is considered to have a very poor interpretability because the unstructured collection of training instances is far from readable, especially if there are many instances.

While interpretability concerns a typical classifier generated by a learning algorithm, transparency (or comprehensibility) refers to whether the principle of the method of constructing a classifier is easy to understand (that is a users' subjective assessment). Therefore, for example, a kNN classifier is scarcely interpretable, but the method itself is transparent because it appeals to the intuition of humans who spontaneously reason from similar cases. Similarly, interpretability of Naïve Bayes can be estimated as not very high, but the transparency of the method is good for example for physicians who find that probabilistic explanations replicate their way of diagnosing, i.e., by summing evidence for or against a disease (Kononenko, 1993).

Our goal is to analyse the drawbacks and advantages (if any) of FE process on the interpretability and transparency of different rule-based and instance-based SL algorithms.

3.10 Putting all together: towards the framework of DM strategy selection

The purpose of studying and developing some pieces of theory background or practical aspects of FE for SL is certainly the *rigor*-related type of research. However, in this thesis we address also the *relevance* issues. With this respect our main distant (faraway) research goal is to contribute to knowledge in the problem of data mining strategy selection for a certain data mining problem. And our particular focus is on different combinations of considered PCA-based FE techniques and SL techniques.

Our aim is to introduce the general framework of KDD system that would incorporate DSS approach to provide help for the user in the selection of the most appropriate data mining strategy for a data set under consideration and to allow mixed-initiative management of automated KDD process (see Figure 13).

A key idea is to apply the meta-learning approach for automatic algorithm selection (see for example Kalousis (2002) for an overview). There exist two contexts of meta-learning. The first one is related to the so-called multi-classifier systems that apply different ensemble techniques (Dietterich, 1997). Their general idea is usually to select one classifier on the dynamic basis taking into account the local performance (for example generalisation accuracy) in the instance space (see Article VII). In the second, the multi-strategy learning applies strategy selection approach which takes into account the classification problem related characteristics (meta-data).

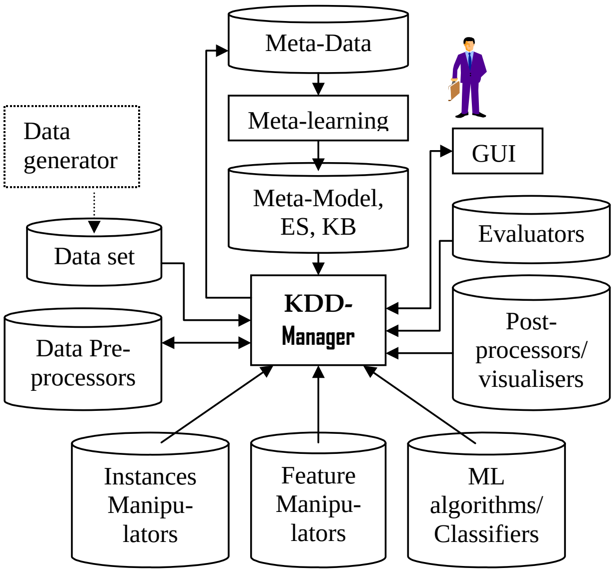


FIGURE 13 DM strategy selection via meta-learning and taking benefit of CI approach (see Article IX for the detailed description)

4 RESEARCH METHODS AND RESEARCH DESIGN

In this chapter research methods and research design of the thesis are considered. First, we introduce our view on DM research in the scope of Information Systems Development (ISD) perspective and consider three basic research approaches used in the study. Then, we focus our discussion on basic approaches for evaluating learned models and for evaluating DM techniques that are used to construct these models.

4.1 DM research in the scope of ISD research methods

In Pechenizkiy *et al.* (2005a) we consider the DM research as a continuous Information Systems Development (ISD) process. We refer to the traditional framework presented by Ives *et al.* (1980) that is widely known and has been used in the classification of Information Systems (IS) research literature. Drawing an analogy to this framework we consider a DM system as a special kind of adaptive information system that processes data and helps to make use of it. Adaptation in this context is important because of the fact that the DM system is often aimed to produce solutions to various real-world problems, and not to a single problem. On the one hand, a DM system is equipped with a number of techniques to be applied for a problem at hand. On the other, there exist a number of different problems, and current research has shown that no single technique can dominate some other technique over all possible data mining problems (Wolpert & MacReady, 1996). Nevertheless, many empirical studies report that a technique or a group of techniques can perform significantly better than any other technique on a certain DM problem or a group of problems (Kiang, 2003). Therefore DM research can be seen as a development process of a DM system aimed at efficient utilization of available DM techniques for solving a current problem.

Focusing on the ISD process, we consider ISD framework of Nunamaker *et al.* (1990-91) adapted to DM artefact development. We discuss three basic groups of IS research methods. Namely, we consider theoretical, constructive and experimental approaches with regard to Nunamaker's framework in the context of DM. We demonstrate how these approaches can be applied iteratively and/or in parallel for the development of an artefact – a DM tool, and contribute to theory creation and theory testing.

Iivari *et al.* (1999) relate development process to the constructive type of research because of their philosophical belief that development always involves creation of some new artefacts – conceptual (models, frameworks) or more technical artefacts (software implementations). The research approach is classified as constructive where scientific knowledge is used to produce either useful systems or methods, including development of prototypes and processes. Iivari *et al.* (1999) argue the importance of constructive research especially for applied disciplines of IS and computer science such as DM.

Nunamaker *et al.* (1990-91, 94) consider system development as a central part of a multi-methodological IS research cycle (Figure 14).

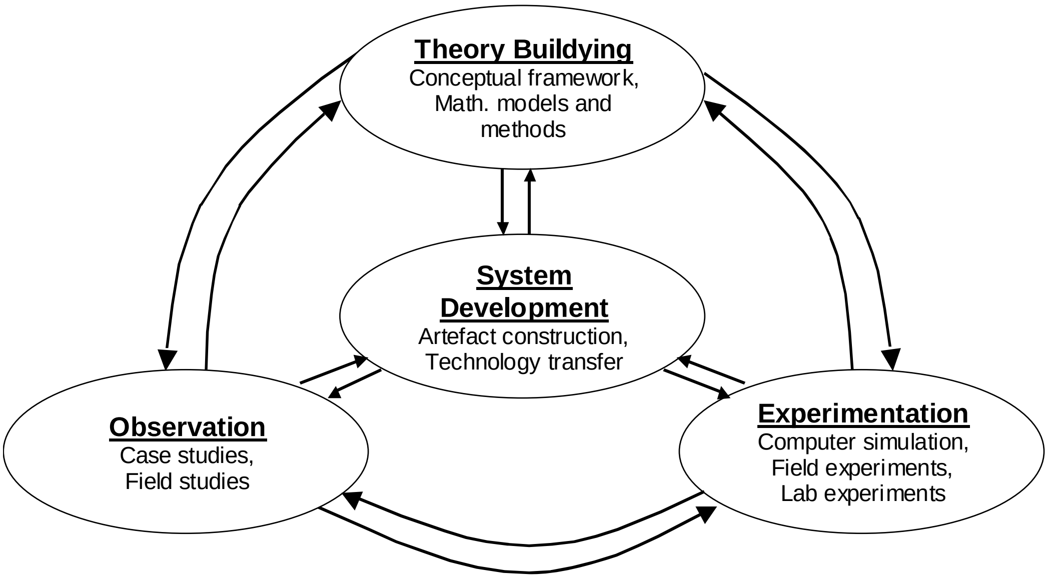


FIGURE 14 A multimethodological approach to the construction of an artefact for DM (adapted from Nunamaker *et al.*, 1990-91, 94)

Theory building involves discovery of new knowledge in the field of study, however it is rarely contributing directly to practice. Nevertheless, the built theory often (if not always) needs to be tested in the real world to show its validity, recognize its limitations and make refinements according to observations made during its application. Therefore, research methods are subdivided into basic and applied research, as naturally both are common for any large project (Nunamaker *et al.*, 1990-91). A proposed theory leads to the development of a prototype system in order to illustrate the theoretical framework, and to test it through experimentation and observation with subsequent refinement of the theory and the prototype in an iterative manner. Such a view presents DM research as a complete, comprehensive and dynamic process. It allows multiple perspectives and flexible choices of methods to be applied during different stages of the research process. Furthermore, following this multimethodological approach a researcher can analyze how the results achieved through different research approaches relate to each other and search for contradictions in the results. It can be expected that such joint use of these approaches will give a better understanding of the introduced research goal and provide a more significant and sophisticated contribution to the knowledge in the area.

4.2 Research methods used in the study

Three basic research approaches are used in this thesis: *the conceptual-theoretical approach*, *the constructive approach*, and *the experimental approach*. These approaches are tightly connected and are applied in parallel. The theoretical background is exploited during the constructive work and the constructions are used for experimentation. The results of the constructive and experimental work are used to refine the theory. Accordingly, several *research methods* are applied.

In the conceptual-theoretical approach, conceptual basics and formalisms of the integration of multiple DM methods in knowledge discovery systems, and especially dynamic integration, are reviewed and discussed. During the constructive part of the research, software that implements the developed theory and allows conduct the experimental study and evaluation is developed. In the experimental part of the research, widely available benchmark databases (artificial and real-world ones) are used to evaluate characteristics of the developed integration approach in order to obtain deeper understanding about its behaviour in different subject domains.

The constructive approach, from the DM research point of view, can be seen as means that helps to manipulate and coordinate integrative work of different DM methods, and to carry out the experimental approach. It is obvious that in order to construct a good artefact we need some background knowledge about artefacts' components (that are basic DM techniques) and their appropriateness for certain data set characteristics. Thus, it is natural that theory-creating research has to be performed, during which the basics of the relevant DM techniques should be elaborated. For these purposes literature survey and review was done. This helped us to understand the background of the problem and to analyse the previous findings in the area.

During the *development process* of our constructive research we used MLC++ (the machine learning library in C++) (Kohavi *et al.*, 1996) and WEKA machine-learning environment in Java (Witten & Frank, 2000). This allowed us to use tested and validated tools as a core/backbone for a new tool. We chose component-based development as it allows each component to be designed, implemented, tested and refined independently. The control over the individual components is easier to organize and the experiments can be more easily performed on the separate components also.

Evaluation process is an essential part of constructive research. Naturally, experimental approach was used to evaluate the prototype. The experimental approach was beneficial for theory testing and theory construction.

Experimental study can be done in the 'field' or in the 'laboratory'. In the first case different approaches are tested on so-called real-world datasets with real users. In the second case systematically controlled experiments can be organized. Controlled experiments sometimes might produce more beneficial results for theory creating, since unlike real world datasets, synthetically generated data allow testing exactly the desired number of characteristics while keeping all the others unchanged.

In the next two sections, the experimental approach and experiment design are considered in more detail.

4.3 Experimental approach

Evaluation process is an essential part of constructive research. By the evaluation of a DM artefact we understand first of all (1) the evaluation of learned models and meta-level models and (2) testing the hypothesis about superiority of one studied technique or a combination of techniques over another one. Some other important issues related to the use of DM artefact are discussed in Pechenizkiy *et al.* (2005a). However, the experimental approach benefits not only the artefact evaluation and theory testing that has been used for artefact construction but also it can contribute to knowledge producing new pieces of theory for selection and/or combination of DM techniques for a given dataset. Meta-learning approaches are one good example of such attempts to contribute to new pieces of theory induction.

4.3.1 Estimating the accuracy of the model learnt

For the purposes of algorithm comparison and selection, as well as for parameter setting, methods of estimating the performance of a set of learned models are needed. The goal of the model selection task is to estimate the generalization performance of a collection of learning algorithms and to select the algorithm with the lowest error estimate (Kohavi, 1995a).

When testing and validating a model, data miners use several techniques. They include sampling, validation, cross-validation, stratification, Monte Carlo methods, division of dataset into training, validating and testing sets etc. The two most essential elements of any experimental design are randomization and experimental control of adjustable variables and restrictions of known factors.

One way to estimate the accuracy is to use the *resubstitution estimate*, in which the model is tested on the same data it was built on (Kohavi, 1995b). Although the resubstitution estimate is a highly optimistic estimate of accuracy it was noted that for large enough samples, and for some algorithms there is no need to look further than the resubstitution estimator (Kohavi, 1995b).

However, the common approach is to use a sample of the available previously classified instances for a training set and the remaining instance – for a test set. Then the training set is used to learn the model, and the test set is used to test it. The major nonparametric statistical methods that follow this methodology are cross-validation, random sampling (Monte Carlo cross-validation), and bootstrapping (Merz, 1998).

In cross-validation (Schaffer, 1993) the examples are randomly split into *v* mutually exclusive partitions (folds) of approximately equal size. A sample is formed by setting aside one of the *v* folds as the test set, and the remaining folds make up the training set. This creates *v* possible samples. As each learned model is formed using one of the *v* training sets, its generalization performance is estimated on the corresponding test partition. S*tratified* cross-validation, where the folds are stratified so that they contain approximately the same proportions of classes as the original dataset, can give better estimation (Kohavi, 1995a). Usually, multiple runs of cross-validation are used for stabilization of the estimations (Kohavi, 1995b).

Random sampling or Monte Carlo cross-validation (Kohavi, 1995a) is a special case of *v*-fold cross-validation where a percentage of training examples (typically 2/3) is randomly placed in the training set, and the remaining examples are placed in the test set. After learning takes place on the training set, generalization performance is estimated on the test set. This whole process is repeated for many training/test splits (usually 30) and the algorithm with the best average generalization performance is selected. Random sampling is used in most experiments throughout this dissertation to evaluate the methods developed and to compare them with the existing ones.

Bootstrapping (Kohavi, 1995a) is the process of sampling with replacement from the available examples to form the training and test partitions. Kohavi (1995a) showed that on average, cross-validation methods are better than bootstrapping, and could be recommended for accuracy estimation and model selection.

The evaluation of an DM technique can be either based on filter paradigm, when evaluation process is independent from a learning algorithm and the most appropriate approach is chosen from available ones according to certain data characteristics before the algorithm starts, or based on wrapper paradigm (Kohavi & John, 1998) that assumes the interaction between the approach selection process and performance of the integrative model. In this thesis the wrapper approach is used in the experimental studies.

4.3.2 Tests of hypotheses

From the theory evaluation as well as from the artefact evaluation point of view, the general principle of evaluation – the new derivation or construct must be better that its best challenger – is applicable for DM as well. 'Goodness' criterion of a built theory or an artefact is multidimensional and sometimes is difficult to define because of mutual dependence between the compromising estimates. However, it is fairly easy to construct a criterion based on such estimates as accuracy (including sensitivity and specificity, and various costs matrices) of a built model and its performance (time and memory resources). On the other hand – it is more difficult or even impossible to include into a criterion such important aspects as interpretability of the artefact's output because estimates of that kind usually are subjective and can be evaluated only by the end-users.

When a new DM technique is compared with some existing technique (competitor) the cross-validation methods are commonly used to estimate their generalization performance. As a rule, it is necessary to determine how significant the observed differences are. The resampled Student's t-test (also known as the resampled paired t-test) is one commonly used tool that has many potential drawbacks (Dietterich, 1998).

For the resampled paired *t*-test, a series of trials (usually 30) is conducted. In each trial, the available sample is randomly divided into a training set and a test set (for example, two thirds and one third of the data correspondingly). Learning algorithms *A* and *B* are both trained on the training set and the resulting classifiers are tested on the test set. Let $p_A^{(i)}$ (respectively $p_B^{(i)}$) be the observed proportion of test examples misclassified by algorithm *A* (respectively *B*) during trial *i*. If we assume that the 30 differences $p^{(i)} = p_A^{(i)} - p_B^{(i)}$ were drawn independently from a normal distribution, then we can apply the Student's *t* test, by computing the statistic

$$t = \frac{\bar{p} \cdot \sqrt{n}}{\sqrt{\frac{\sum_{i=1}^{n} (p^{(i)} - \bar{p})^2}{n-1}}}, \tag{12}$$

where $\bar{p} = \frac{1}{n} \cdot \sum_{i=1}^{n} p^{(i)}$, and *n* is the number of trials. Under the null hypothesis, this statistic has a *t* distribution with *n*-1 degrees of freedom. For example for 30 trials, the null hypothesis can be rejected if $|t| > t_{29,0.975} = 2.04523$.

Usually, neither independence of algorithms ($p_A^{(i)}$ and $p_B^{(i)}$) nor independence of each evaluation from the others (because of overlapping of the training sets in the trials) is guaranteed. Recent studies (Dietterich, 1998; Salzberg, 1999) have shown that the resampled t-test and other commonly used significance tests have an unacceptably high probability of detecting a difference in generalization performance when no difference exists (Type 1 error). This is primarily due to the nature of the sampling process in the experimental design and the number of examples available.

The McNemar's test, and the test for the difference of two proportions are claimed to have acceptable Type 1 error (Dietterich, 1998). Nevertheless, in Tsymbal (2002), the same results were obtained with all the tests, sometimes with different levels of significance. However, although no single procedure for comparing learning methods based on limited data satisfies all the constraints, each of them provides the approximate confidence intervals that can be used in interpreting experimental comparisons of learning techniques (Mitchel, 1997).

4.4 Experimental design

In this section, the most common experimental settings used throughout this study are described. More detailed experimental settings can be found in the corresponding section of the related article included in the thesis.

To compare the developed algorithms and the existing ones, for each data set, 30 or 70 test runs were made. In each test run a data set was first split into the training set, the validation set, and the test set by stratified random sampling. Each time 70 percent of the instances were included in the training set. The other 30 percent were used for the test set. When the validation set is used (for example in the iterative refinement of the ensemble) 60 percent of the instances were included in the training set, and the other 40 percent were divided into the validation and test sets of approximately equal size. The test set was used for the final estimation of the ensemble accuracy.

When needed, the values of continuous features are discretized dividing the interval of the values of the feature into intervals with equal length. The whole experimental environment was implemented first within the MLC++ framework. The results described in Article I, Article VII and Article VIII were achieved using that experimental environment. For our further studies we implemented the experimental environment within WEKA machine-learning environment in Java (WEKA 3, 2004).

The datasets used in the experiments were taken from the University of California at Irvine Machine Learning Repository (Blake & Merz., 1998), except for the *Acute Abdominal Pain* (AAP) datasets provided by Laboratory for System Design, Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia and Theoretical Surgery Unit, Dept. of General and Trauma Surgery, Heinrich-Heine University Düsseldorf, Germany (Zorman *et al.*, 2001), and microbiology datasets *Antibioticograms* provided by N. N. Burdenko Institute of Neurosurgery, Russian Academy of Medical Sciences, Moscow, Russia. The short summary and the description of all the datasets used throughout the different experiments can be found in Appendix A.

In the experiments, four FE techniques considered in Section 2.5: conventional PCA, random projections, the parametric and nonparametric class-conditional FE techniques; three supervised learning techniques considered in Section 2.2: Naïve Bayes, *k* Nearest Neighbour, and C4.5 Decision Tree; three integration techniques considered in Section 2.3: dynamic selection, dynamic voting, dynamic voting with selection; and four sampling techniques considered in Section 2.6: random, stratified random, and *kd*-tree based selective sampling (with and without stratification) were used. In the evaluation of different DM strategies composed by the above mentioned techniques we were interested in their generalization accuracy, number of features required to construct a model, and the time taken to construct and test a model.

5 RESEARCH RESULTS: SUMMARY OF THE INCLUDED ARTICLES

This chapter presents a brief discussion of each article included in the thesis and discusses the main findings of each corresponding article. Generally, each included article addresses the research problem(s) presented in the corresponding section(s) of Chapter 3.

5.1 "Eigenvector-based feature extraction for classification"

Reference: Tsymbal, A., Puuronen, S., Pechenizkiy, M., Baumgarten, M. & Patterson D. 2002. Eigenvector-based Feature Extraction for Classification. In: S.M. Haller, G. Simmons (Eds.), Proceedings 15th International FLAIRS Conference on Artificial Intelligence, FL, USA: AAAI Press, 354-358.

PCA-based FE techniques are widely used for classification problems, though they generally do not take into account the class information and are based solely on inputs. Although this approach can be of great help in unsupervised learning, there is no guarantee that the new axes are consistent with the discriminatory features in a classification problem.

This paper shows the importance of the use of class information in FE for SL and inappropriateness of conventional PCA to FE for SL. We considered two class-conditional eigenvector-based approaches for FE described in Subsection 2.5.3. We compared the two approaches with each other, with conventional PCA, and with plain nearest neighbor classification without FE.

First, a series of experiments were conducted to select the best α and k coefficients for the nonparametric approach. The parameter α was selected from the set of 9 values: $\alpha \in \{1/20, 1/10, 1/5, 1/3, 1, 3, 5, 10, 20\}$, and the number of nearest neighbors n from the set of 8 values: $k = 2^i - 1,\ i = 1,\dots,8$, $k \in \{1, 3, 7, 15, 31, 63, 127, 255\}$. The parameters were selected on the wrapper-like basis, optimizing the classification accuracy. For some data sets, for example LED and LED17, selection of the best parameters did not give almost any improvement in comparison with the ones considered by Fukunaga (1990) α =1 and k=3, and the classification accuracy varied within the range of one percent. It is necessary to note that the selection of the α and n parameters changed the ranking of the three feature extraction approaches from the accuracy point of view only on two data sets, thus demonstrating that the nonparametric approach is robust with regard to the built-in parameters. However, for some data sets the selection of the parameters had a significant positive effect on the classification accuracy. For example, on the MONK-2 data set, accuracy is 0.796 when α =1 and k=3, but it reaches 0.974 when α =20 and k=63.

The nonparametric approach had the best accuracy on average. Also, the nonparametric approach performed much better on the categorical data, improving the accuracy of the other approaches for this selection of the data sets. However, further research is necessary to check this finding. The parametric approach was quite unstable, and not robust to different data sets' characteristics. Conventional PCA was the worst FE technique on average. Classification without FE was clearly the worst. This shows the so-called "curse of dimensionality" and necessity of FE.

Thus, the experimental results supported our expectations. Still, it is necessary to note that each feature extraction technique was significantly worse than all the other techniques at least on one data set (for example, the Heart data set for the nonparametric approach), and it is a question for further research to define the dependencies between the characteristics of a data set and the type and parameters of the feature extraction approach best suited for it.

5.2 "PCA-based Feature Transformations for Classification: Issues in Medical Diagnostics"

Reference: Pechinizkiy, M., Tsymbal, A. & Puuronen, S. 2004. PCA-based Feature Transformations for Classification: Issues in Medical Diagnostics, In: R. Long *et al.* (Eds.), Proceedings of 17th IEEE Symposium on Computer-Based Medical Systems CBMS'2004, Bethesda, MD: IEEE CS Press, 535-540.

Current electronic data repositories, especially in medical domains, contain enormous amounts of data including also currently unknown and potentially interesting patterns and relations that can be uncovered using knowledge discovery and DM methods. Inductive learning systems were successfully applied in a number of medical domains, for example in the localization of a primary tumor, prognostics of recurrence of breast cancer, diagnosis of thyroid diseases, and rheumatology.

However, researchers and practitioners realize that the effective use of these inductive learning systems requires data preprocessing before applying a learning algorithm. This is especially important for multidimensional heterogeneous data, consisting of a large number of features of different types. If a subset of irrelevant and/or redundant features can be recognized and eliminated from the data then feature selection techniques may work well. Unfortunately this is not always easy and sometimes not even possible. This is because a feature subset may be useful in one part of the instance space, and at the same time be useless or even misleading in another part of it. That is why the transformation of the given representation before weighting the features is often preferable.

FE is often seen as a dimensionality reduction technique. An alternative is to see FE as a useful transformation that leads to representation space improvement due to elimination of correlated and uninformative features and construction of uncorrelated and informative ones. However, when the number of original features is relatively small, some new features produced by means of FE process may give additional value for the set of original ones.

In this paper we studied advantages of using either extracted features or both original and extracted features for SL.

We elaborated a test bench with a collection of medical data sets taken from the UCI machine learning repository and on three data sets of cases of acute abdominal pain to conduct experiments with the considered FE techniques (Section 2.5) and the kNN-classifier. We evaluated four combinations of kNN with the considered PCA-based FE techniques (conventional PCA, ranked PCA, parametric eigenvalue-based approach, and nonparametric eigenvalue-based approach). Then we evaluated the results of these four combinations against the best wrapper procedure. After that we compared the same combinations to find out whether the replacement of the initial features by the extracted ones is better than their superposition.

Our experimental results showed that for *Diabetes* and *Thyroid* data sets none of the feature transformation techniques can improve the work of a plain 3NN classifier. For *Heart* and *Cancer* data sets 3NN achieves the highest accuracy results when the new features extracted by the parametric approach are used instead of the original ones. And for *Liver* data set the best results are achieved when the feature extracted by the parametric approach is used together with the original ones. KMO criterion (Section 3.8) being successfully used in factor analysis is not useful when deciding whether FE will improve the representation space for SL. Although for every data set KMO was higher than 0.5, the principal components, when used instead of the original features, resulted in lower accuracy of 3NN classifier, and when used together with the original features, never improved the classification accuracy.

We also discussed some interpretability issues of the new extracted features. We argued that, although when applied for rule-like approaches (association rules), the interpretation of rules with respect to the initial (original) features may be difficult or even impossible, for case-based approaches (nearest neighbor) the comparison by analogy may be easier. Additionally we discussed whether the transformation formulas of principal components may provide useful information for interpretability of results; and whether the interpretability can be improved with the help of a new feature space rotation and back-transformation (where such operations are appropriate and applicable).

5.3 "On Combining Principal Components with Fisher's Linear Discriminants for Supervised Learning"

Reference: Pechenizkiy, M., Tsymbal, A. & Puuronen, S. 2005. On Combining Principal Components with Fisher's Linear Discriminants for Supervised Learning. (submitted to) Special Issue of Foundations of Computing and Decision Sciences "Data Mining and Knowledge Discovery" (as extended version of Pechenizkiy *et al.*, 2005e).

In this paper, principal component analysis (PCA), parametric feature extraction (FE) based on Fisher's linear discriminant analysis (LDA), and their combination as means of dimensionality reduction are analyzed with respect to the performance of classifier. Three commonly used classifiers are taken for analysis: kNN, Naïve Bayes and C4.5 decision tree. Recently, it has been argued that it is extremely important to use class information in FE for supervised learning (SL). However, LDA-based FE, although using class information, has a serious shortcoming due to its parametric nature. Namely, the number of extracted components cannot be more that the number of classes minus one. Besides, as it can be concluded from its name, LDA works mostly for linearly separable classes only.

In this paper we study whether it is possible to overcome these shortcomings by adding the most significant principal components to the set of features extracted with LDA. In experiments on 21 benchmark datasets from UCI repository these two approaches (PCA and LDA) are compared with each other, and with their combination for each classifier.

Our results show that such a combination approach has certain potential, especially when applied for C4.5 decision tree learning. However, from the practical point of view the combination approach cannot be recommended for Naïve Bayes since its behavior is very unstable on different datasets. Presumably, additional feature selection would be useful for Naïve Bayes and kNN from the combined set of features, which is implicitly done with C4.5.

5.4 "The Impact of the Feature Extraction on the Performance of a Classifier: kNN, Naïve Bayes and C4.5"

Reference: Pechenizkiy, M. 2005. The Impact of the Feature Extraction on the Performance of a Classifier: kNN, Naïve Bayes and C4.5. In: B.Kegl & G.Lapalme (Eds.): Proceedings of 18th CSCSI Conference on Artificial Intelligence AI'05, LNAI 3501, Heidelberg: Springer-Verlag, 268-279.

In this paper we analyzed FE from two different perspectives. The first one is related to the "curse of dimensionality" problem and the necessity of dimensionality reduction (see Section 2.4). The second perspective comes from the assumption that in many data sets to be processed some individual features, being irrelevant or indirectly relevant for the purpose of analysis, form poor problem representation space. Corresponding ideas of constructive induction that assume the improvement of problem representation before application of any learning technique are presented (see Section 2.6).

FE accounts for both of the perspectives, and therefore, FE, when applied either on data sets with high dimensionality or on data sets including indirectly relevant features, can improve the performance of a classifier.

One main hypothesis is that different FE techniques might have different effects for different classifiers.

We conducted the experiments with four different types of FE techniques: PCA, Random Projection, and two class-conditional approaches to FE, and with three SL algorithms: the nearest neighbour classification, Naïve Bayes, and C4.5 decision tree learning, analyzing the impact of FE techniques on the classification performance on 20 UCI datasets.

In this paper, the experimental results show that for many data sets FE does increase classification accuracy. Still, we could see from the results that there is no best FE technique among the considered ones, and it is hard to say which one is the best for a certain classifier and/or for a certain problem, however according to the experimental results some major trends can be recognized.

Class-conditional approaches (and especially nonparametric approach) were often the best ones. This indicated the importance of taking into account class information and not relying only on the distribution of variance in the data. At the same time it is important to notice that the parametric FE was very often the worst, and for 3NN and C4.5 the parametric FE was the worst more often than RP. Such results highlight the very unstable behavior of parametric FE. One possibility to improve the parametric FE would be to combine it with PCA or a feature selection approach in a way that a few principal components or the components most useful for classification features are added to those extracted by the parametric approach. We experimentally evaluated this idea later in Article III; the results showed that parametric FE produces more stable results when its extracted features are combined with few principal components for SL.

Although it is logical to assume that RP should have more success in applications where the distances between the original data points are meaningful and/or for such learning algorithms that use distances between the data points, our results show that this is not necessarily true. However, data sets in our experiments have 48 features at most and RP is usually applied for problems with much higher dimensionality.

The main conclusion of the paper is that FE techniques are powerful tools that can significantly increase the classification accuracy producing better representation spaces or resolving the problem of "the curse of dimensionality". However, when applied blindly, FE may have no effect for the further classification or can even deteriorate the classification accuracy.

5.5 "Local Dimensionality Reduction within Natural Clusters for Medical Data Analysis"

Reference: Pechenizkiy, M., Tsymbal, A., Puuronen, S. 2005. Supervised Learning and Local Dimensionality Reduction within Natural Clusters: Biomedical Data Analysis, (submitted to) IEEE Transactions on Information Technology in Biomedicine, Special Post-conference Issue "Mining Biomedical Data" (as extended version of Pechenizkiy *et al.*, 2005c)

Inductive learning systems have been successfully applied in a number of medical domains. Nevertheless, the effective use of these systems requires data preprocessing before applying a learning algorithm. It is especially important for multidimensional heterogeneous data, presented by a large number of features of different types. Dimensionality reduction is one commonly applied approach. The goal of this paper was to study the impact of "natural" clustering on dimensionality reduction for classification. We compared several DM strategies that apply dimensionality reduction by means of FE or feature selection for subsequent SL with the selected part of real clinical database trying to construct data models that would help in the prediction of antibiotic resistance and in understanding its development.

Each instance of the data used in our analysis represents one *sensitivity test* and contains the features related to *pathogen* that is isolated during the microbe identification analysis, *antibiotic* that is used in the sensitivity test and the *result* of the sensitivity test (sensitive, resistant, or intermediate). The information about sensitivity analysis is connected with a *patient*, his/her demographical data (*sex*, *age*) and hospitalization in the Institute (*main department*, whether the test was taken while patient was in *ICU (*Intensive Care Unit*)*, *days spent* in the hospital before, etc.). We introduced grouping features for pathogens and antibiotics so that 17 pathogens and 39 antibiotics were combined into 6 and 15 groups respectively. Thus, each instance had 28 features that included information corresponding to a single sensitivity test augmented with data concerning the type of the antibiotic used and the isolated pathogen, and clinical features of the patient and his/her demographics, and the microbiology test result as the class attribute. The data is relatively high dimensional and heterogeneous; heterogeneity is presented by a number of contextual (environmental) features. In this study we applied natural clustering aimed to use contextual features for splitting a real clinical dataset into more homogeneous clusters in order to construct local data models that would help in better prediction of antibiotic resistance. Semantically, the *sensitivity* concept is related first of all to the *pathogen* and *antibiotic* concepts. For our study binary features that describe the pathogen grouping were selected as prior environmental features, and they were used for hierarchical natural clustering (the hierarchy was introduced by the grouping of the features). So, the whole dataset was divided into two nearly equal natural clusters: *gram+* and *gram–*. Then, the *gram+* cluster divided into the *staphylococcus* and *enterococcus* clusters, and *gram–* cluster divided into the *enterobacteria* and *nonfermentes* clusters.

We analyzed experimentally whether local dimensionality reduction within "natural" clusters is better than global search for a better feature space for classification in terms of performance.

In our experimental study we applied k-nearest neighbor classification (kNN) to build antibiotic sensitivity prediction models. We applied three different wrapper-based sequential FS techniques and three PCA-based FE techniques globally and locally and analyzed their impact on the performance of kNN classifier.

The results of our experiments showed that natural clustering is very effective and efficient approach to cope with complex heterogeneous datasets, and that the proper selection of a local FE technique can lead to significant increase of predictive accuracy in comparison with the global kNN with or without FE. The amount of features extracted or selected locally is always smaller than that in the global space, which shows the usefulness of natural clustering in coping with data heterogeneity.

5.6 "The Impact of Sample Reduction on PCA-based Feature Extraction for Naïve Bayes Classification"

Reference: Pechenizkiy, M., Puuronen, S. & Tsymbal, A. 2006. The Impact of Sample Reduction on PCA-based Feature Extraction for Supervised Learning. (to appear) In: Proceedings of the 21st ACM Symposium on Applied Computing (SAC'06, Data Mining Track), ACM Press.

When a data set contains a huge number of instances, some sampling approach is applied to address the computational complexity of FE and classification processes. The focus of this paper is within the study of sample reduction effect on FE techniques with regard to the classification performance.

The main goal of this paper is to show the impact of sample reduction on the process of FE for classification. In our study we analyzed the conventional

Principal Component Analysis (PCA) and two eigenvector-based approaches that take into account class information (Section 2.5). The experiments were conducted on ten UCI data sets, using four different strategies to select samples: (1) random sampling, (2) stratified random sampling, (3) *kd*-tree based selective sampling, and (4) stratified sampling with *kd*-tree based selection.

The experimental results of our study showed that the type of sampling approach is not important when the selected sample size is relatively large. However, it is important to take into account both class information and information about data distribution when the sample size to be selected is small.

The *kd*-tree based sampling has very similar effect to stratified random sampling, although different in nature.

Comparing the results related to the four sampling strategies we can conclude that no matter which one of the four sampling strategies is used, if sample size p is small, $p \approx 10\%$, then SL without FE yields the most accurate results; if sample size $p \geq 20\%$, then nonparametric class-conditional FE (*NPAR*) outperforms other methods; and if sample size $p \geq 30\%$, *NPAR* outperforms other methods even if they use 100% of the sample. The best p for *NPAR* depends on sampling method: for random and stratified $p = 70\%$, for *kd*-tree $p = 80\%$, and for stratified + *kd*-tree $p = 60\%$. PCA is the worst technique when applied on a small sample size, especially when stratification or *kd*-tree indexing is used.

Generally, all sampling strategies have similar effect on final classification accuracy of NB for $p > 30\%$. The significant difference in accuracy is within $10\% \leq p \leq 30\%$. The intuitive explanation for this is that when taking a very large proportion of the sample, it does not matter which strategy is used since most of the selected instances are likely to be the same ones (maybe chosen in different orders). However, the smaller the portion of the sample, the more important it is how the instances are selected.

5.7 "Feature extraction for dynamic integration of classifiers"

Reference: Pechenizkiy, M., Tsymbal, A., Puuronen, S. & Patterson, D. 2005. Feature Extraction for Dynamic Integration of Classifiers, (submitted to) Fundamenta Informaticae, IOS Press (as extended version of Tsymbal *et al.*, 2003).

Recent research has shown the integration of multiple classifiers to be one of the most important directions in machine learning and DM. It was shown that, for an ensemble to be successful, it should consist of accurate and diverse base classifiers. However, it is also important that the integration procedure in the ensemble should properly utilize the ensemble diversity. In this paper, we present an algorithm for the dynamic integration of classifiers in the space of extracted features (FEDIC). It is based on the technique of dynamic integration, in which local accuracy estimates are calculated for each base classifier of an ensemble, in the neighbourhood of a new instance to be processed. Generally, the whole space of original features is used to find the neighbourhood of a new instance for local accuracy estimates in dynamic integration. We propose to use FE in order to cope with the curse of dimensionality in the dynamic integration of classifiers. We consider classical principal component analysis and two eigenvector-based supervised FE methods that take into account class information and their application to the dynamic selection, dynamic voting and dynamic voting with selection integration techniques (DS, DV and DVS). Experimental results show that, on some data sets, the use of FEDIC leads to significantly higher ensemble accuracies than the use of plain dynamic integration in the space of original features. As a rule, FEDIC outperforms plain dynamic integration on data sets, on which both dynamic integration works well (it outperforms static integration), and considered FE techniques are able to successfully extract relevant features.

Our main hypothesis was that with data sets, where FE improves classification accuracy when employing a single classifier (such as kNN), it would also improve classification accuracy when a dynamic integration approach is employed. Conversely, with data sets, where FE decreases (or has no effect on) classification accuracy with the use of a single classifier, FE will also decrease (or will have no effect on) classification accuracy when employing a dynamic integration approach.

The results supported our hypothesis and showed that the proposed FEDIC algorithm outperforms the dynamic schemes on plain features only on those data sets in which FE for classification with a single classifier provides better results than classification on plain features. When we analyzed this dependency further, we came to a conclusion that FE influenced the accuracy of dynamic integration in most cases in the same manner as FE influenced the accuracy of base classifiers.

We conducted further experimental analyses on those data sets on which FEDIC was found to produce significantly more accurate results than DIC. For each data set we compared the behavior of conventional PCA versus class-conditional approaches with respect to DS, DV and DVS, and vice versa, the behavior of integration strategies with respect to FE techniques.

A number of meta-features that are used to search for the nearest neighbor were compared with respect to the cases with and without FE. We then analyzed how FE techniques improve the neighborhood of each data set on average and found strong correlation between these results with the generalized accuracy results.

5.8 “Feature extraction for classification in knowledge discovery systems”

Reference: Pechenizkiy, M., Puuronen, S. & Tsymbal, A. 2003. Feature extraction for classification in knowledge discovery systems, In: V.Palade, R.J.Howlett, L.C.Jain (Eds.), Proceedings of 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems KES’2003, Lecture Notes in Artificial Intelligence, Vol.2773, Heidelberg: Springer-Verlag, 526-532.

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 next-generation DM systems to manage knowledge discovery applications. These systems should be able to discover knowledge by combining several available techniques, and provide a more automatic environment, or an application envelope, to surround this highly sophisticated DM engine.

In this paper we considered a decision support system (DSS) approach that is based on the methodology used in expert systems. The approach is aimed to combine FE techniques with different classification tasks. The main goal of such system is to automate as much as possible the selection of the most suitable FE approach for a certain classification task on a given data set according to a set of criteria.

Although there is a huge number of FE methods (that apply linear or nonlinear processing, and are applied globally or locally), currently, as far as we know, there is no FE technique that would be the best for all data sets in the classification task. Thus the adaptive selection of the most suitable FE technique for a given data set is a real challenge. Unfortunately, there does not exist canonical knowledge, a perfect mathematical model, or any relevant tool to select the best extraction technique. Instead, a volume of accumulated empirical findings, some trends, and some dependencies have been discovered.

In order to help to manage the DM process, recommending the best-suited FE method and a classifier for a given data set, we proposed to take benefit of experimental approach for relevant knowledge discovery. Discovered during the experimental research, pieces of knowledge in the form of association rules may save a great amount of time when selecting or at least initialising methods’ parameters in a proper way, and moreover when selecting/recommending the most appropriate combination(s) of FE and classification methods.

Thus, potentially, it might be possible to reach a performance close to the *wrapper* type approach, when actually using the *filter* paradigm, because of the selection of methods and their parameters according to a certain set of criteria in advance.

During the pilot studies we did not find a simple correlation-based criterion to separate the situations where a FE technique would be beneficial for the classification. Nevertheless, we found out that there exists a trend between the correlation ratio in a data set and the threshold level used in every FE method to address the amount of variation in the data set explained by the selected extracted features. This finding helps in the selection of the initial threshold value as a starting point in the search for the optimal threshold value. However, further research and experiments are required to verify these findings.

5.9 "Data mining strategy selection via empirical and constructive induction"

Reference: Pechenizkiy, M. 2005 Data mining strategy selection via empirical and constructive induction. In: M.H. Hamza (Ed.) Proceedings of the IASTED International Conference on Databases and Applications DBA'05, Calgary: ACTA Press, 59-64.

Recently, several meta-learning approaches have been applied for automatic technique selection by several researchers but with little success. The goal of this paper was (1) to critically analyze such approaches, (2) to consider their main limitations, (3) to discuss why they were unsuccessful and (4) to suggest the ways for their improvement. We introduce a general framework for DM strategy selection via empirical and constructive induction, which is central to our analysis.

Our aim in proposing this framework was to contribute to knowledge in the problem of DM strategy selection for a certain DM problem at hand. We proposed a DSS approach in the framework to recommend a DM strategy rather than a classifier or any other ML algorithm. And the important difference here is that constituting a DM strategy the system searches for the most appropriate ML algorithm with respect to the most suitable data representation (for this algorithm). We believe that a deeper analysis of a limited set of DM techniques (particularly, FE techniques and classifiers) of both the theoretical and experimental levels is a more beneficial approach than application of the meta-learning approach only to the whole range of machine learning techniques at once. Combining theoretical and (semiautomatic) experimental approaches requires the integration of knowledge produced by a human-expert and the meta-learning approach.

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

We considered pairwise comparison of classifiers of the meta-level as more beneficial than 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, and produces more specific characterizations.

With respect to meta-model construction we recommended 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 (for example due to time-/cost-consuming processing), the decision tree can produce nothing but probabilistic reasoning. Decision rules, on the contrary, 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. Thus the decision rules are a more stable approach to meta-learning rather than the decision trees.

We considered the possibility to conduct experiments on synthetically generated datasets that 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 fixed. Beside this, experiments on synthetic datasets allow producing additional instances for the meta-dataset.

5.10 About the joint articles

The present introductory part and Article IV (Pechenizkiy, 2005b) and Article IX (Pechenizkiy, 2005a) have been written solely by the author.

The author of this thesis is the principal author of Article II (Pechenizkiy *et al.*, 2004), Article III (Pechenizkiy *et al.*, 2005d), Article V (Pechenizkiy *et al.*, 2005e), Article VI (Pechenizkiy et al., 2006), Article VII (Pechenizkiy *et al.*, 2005c), and Article VIII (Pechenizkiy *et al.*, 2003). Article I (Tsymbal *et al.*, 2002) has been written in close collaboration by the authors. All the articles included have been refereed by at least two international reviewers and published. All the articles except Article II are full-paper refereed and the Article II is extended abstract refereed. Article I, Article IV, Article VIII, and Article IX, and earlier versions of Article III and Article V have been presented by the author personally at the corresponding conferences.

The developed software prototype within WEKA machine learning library in Java for the experimental studies, and some of the contents of experimental sections in the included articles also represent the independent work done by the author. Analysis of background and review of related work in the included joint papers (for example Section 3 in Article VII) were also done mainly by the author.

6 CONCLUSIONS

FE is an important step in DM/KDD process that can be beneficial for SL in terms of classification accuracy and time complexity (see for example Article I) of model learning and new instances classification.

FE can be considered as a dimensionality reduction technique as well as a technique for construction of better representation space for further supervised learning. FE can improve classification accuracy of a model produced by a learner even for datasets having relatively small number of features.

In this chapter we briefly summarize the main contributions of the thesis with regard to rigor and relevance of accomplished research study, discuss its limitations and overview the directions for future work and finally present the challenges of further research.

6.1 Contributions of the thesis

This thesis contributes to the problem of DM methods integration in the KDD process. All contributions of the thesis are summarized with respect to the stated research question (RQ) in Chapter 3:

RQ 1: How important is it to use class information in the FE process? (Section 3.1)

RQ 2: Is FE a data- or hypothesis-driven constructive induction? (Section 3.2)

RQ 3: Is FE for dynamic integration of base-level classifiers useful in a similar way as for a single base-level classifier? (Section 3.3)

RQ 4: Which features – original, extracted or both – are useful for SL? (Section 3.4)

RQ 5: How many extracted features are useful for SL? (Section 3.5)

RQ 6: How to cope with the presence of contextual features in data, and data heterogeneity? (Section 3.6)

RQ 7: What is the effect of sample reduction on the performance of FE for SL? (Section 3.7)

RQ 8: When is FE useful for SL? (Section 3.8)

RQ 9: Interpretability of the extracted features. (Section 3.9)

RQ 10: How to make a decision about the selection of the appropriate DM strategy (particularly, the selection of FE and SL techniques) for a problem at consideration?

The list of the main contributions of the thesis is also divided into two parts. First, the contributions related to the more theory-based results, and then, the contributions related to the use of FE for SL in a knowledge discovery system (KDS) are considered. We denote contribution related to research question RQ*i* as CRQ*i*. We provide also the reference at every point to the corresponding article included in the collection in the thesis.

6.1.1. Contributions to the theory

The results of our experimental studies showed that:

CRQ 1: Use of class information in FE process is crucial for many datasets. Consequently, class-conditional FE can result in better classification accuracy of a learning model whereas solely variance-based FE has no effect on or deteriorates the accuracy. (Articles I and IV)

CRQ 2: Ranking of different FE techniques in line with the corresponding accuracy results of a SL technique can vary a lot for different datasets. And different FE techniques behave also in a different way when integrated with different SL techniques. Thus, FE process should correspond both to dataset characteristics and the type of SL that follows FE process. (Article IV)

CRQ 3: FE can improve dynamic integration of classifiers for those datasets where FE improves accuracy of an instance-based (such as Nearest Neighbour) classifier. (Article VII)

CRQ 4: Combination of original features with extracted features can be beneficial for SL on some datasets especially when tree-based inducers like C4.5 are used for classification. (Article II). Similarly, combining linear discriminants with few principal components may result in better classification accuracy (compared with the use of either of these approaches) when C4.5 is used. However this combination is not beneficial for Naïve Bayes classifier and it results in very unstable behaviour on different datasets. (Article III)

6.1.2. Contributions to the practice (use) of FE for SL in a KDS

First, it might be valuable to remind that in many experimental studies accomplished during the work on this thesis besides artificial and benchmark datasets, we used real world datasets from medical and microbiology domain areas in order to (1) validate our findings also with dirty real datasets and (2) contribute to the domain area, primarily, to improve the classification accuracy.

The results of our experimental studies showed that:

CRQ 5: The appropriate threshold values used in FE process to account the variance explained by (selected) extracted features varies a lot from one dataset to other. (Article I and VIII)

CRQ 6: Natural clustering is a very efficient approach in DM that allows building local FE and SL models, which outperform corresponding global models in classification accuracy using less number of features for learning (due to utilizing some background knowledge). (Article V)

CRQ 7: Training sample reduction affects the performance of SL with FE rather differently. In general, nonparametric FE results in similar or better accuracy results of a classifier with smaller number of training instances than parametric FE. Our results showed that when the proportion of training instances used to build the FE and the learning model is relatively small it is important to use an adequate sample reduction technique to select more representative instances for the FE process. (Article VI)

CRQ 8: Our preliminary experimental study show that, in general, it is hard to predict ahead when (and for which type of dataset) FE might be useful to apply with regard to SL, and which extracted features and how many of them should be used in supervised learning. (Article VIII)

We presented our vision of interpretability of results and transparency of learning process with regard to FE as transformation of original space.

CRQ 9: Our analysis shows that depending on the kind of data, the meaning of the original features and the problem at consideration and the supervised learning technique, FE can be both beneficial and harmful with these respects. (Article II)

The results and conclusions from the experimental studies and further conceptual-analytic research resulted in:

CRQ 10: the construction of a general framework for the selection of the most appropriate DM strategy according to the knowledge about behaviour (use) of DM techniques and their combinations (that can constitute a DM strategy) on different kinds of datasets. (Article IX)

6.2 Limitations and future work

This section is aimed to highlight some known limitations of the study and to name the main aspects of future work.

6.2.1. Limitations

In the thesis we considered a limited set of FE and SL techniques; and a limited set of data characteristics and method's parameters has been analysed.

FE techniques like PCA transformation are crippled by their reliance on second-order statistics. Though uncorrelated the principal components can be statistically highly dependent (Hyvärinen *et al.,* 2001). Independent component analysis (ICA) that can be seen as an extension to PCA (but use some form of higher-order statistics, which means information not contained in a covariance matrix) accounts for this problem.

If the data components have non-linear dependencies, linear feature transformation will require a larger dimensional representation than would be found by a non-linear technique. There exists a number of non-linear implementations of PCA, (see for example Oja, 1997). These and many other existing FE techniques have not been considered in this thesis; however the same research design can be applied to these groups of techniques.

Most of the study is based on experimental type of research supported by constructive and theoretic approaches. We believe that stronger connections to theoretical background of FE and SL techniques could help to make more significant contribution to the field.

6.2.2. Future work

We see several directions for further research. From the experimental setting side - data sets with significantly higher number of features could be analysed. With this respect further analysis of random projections as means of FE as a pre-processing step for FE may bring interesting findings. Significantly more experiments should be performed on the synthetically generated data sets with predefined characteristics.

While in this thesis mainly accuracy of the approaches was analysed, in the further studies it would be interesting to estimate the algorithmic complexity of different schemes more precisely. Another important and interesting work is to study further the effect of FE on transparency of SL process and interpretability of SL outcomes.

6.3 Further challenges

In this section we take the risk to guess the further main interests and challenges with respect to the topic of the thesis. Our strong belief is that the relevance of DM research should be taken more seriously so that rigor and relevance of research are well-balanced.

From a practical point of view it is important to provide useful tools for DM practitioners. Therefore, the most challenging goals of further research in the area are likely to be related to construction of decision support system for DM strategy recommendation. The application of the meta-learning approach for the discovery of pieces of knowledge about behaviour of different DM strategies on different types of data (Pechenizkiy, 2005b) perhaps would be the first step in addressing this challenge. However, knowledge management issues including knowledge representation, organization, storage, and continuous distribution, integration (from multiple types of sources: DM experts and practitioners, results from laboratory experiments on synthetic datasets and from field experiments on real-world problems) and refinement processes will naturally appear. Besides this, research and business communities, or similar KDSs themselves can organize different so-called trusted networks, where participants are motivated to share their knowledge. We tried to highlight these challenges in Pechenizkiy *et al*. (2005b).

REFERENCES

Achlioptas, D. 2001. Database-friendly random projections. In: P. Buneman (Ed.), Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, New York: ACM Press, 274-281.

Aha, D., Kibler, D. & Albert, M. 1991. Instance-based learning algorithms. Machine Learning 6, 37-66.

Aivazyan, S. 1989. Applied statistics: classification and dimension reduction. Moscow: Finance and Statistics.

Aladjem, M. 1994. Multiclass discriminant mappings. Signal Processing 35(1), 1-18.

Almoy, T. 1996. A simulation study on the comparison of prediction methods when only a few components are relevant. Computational Statistics and Data Analysis 21(1), 87-107.

Arciszewski, T., Michalski, R. & Wnek, J. 1995 Constructive Induction: the Key to Design Creativity. In Proceedings of the 3rd International Round-Table Conference on Computational Models of Creative Design, Queensland, Australia, 397-425.

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

Bingham, E. & Mannila, H. 2001. Random projection in dimensionality reduction: applications to image and text data. Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, San Francisco, California

Blake, C. & Merz, C. 1998. UCI repository of machine learning databases. Dept. of Information and Computer Science, University of California, Irvine, CA. http://www.ics.uci.edu/~mlearn/MLRepository.html

Bloedorn, E., Wnek, J. & Michalski, R. 1993. Multistrategy Constructive Induction: AQ17-MCI, Reports of the Machine Learning and Inference Laboratory, MLI 93-4, School of Information Technology and Engineering, George Mason University.

Breiman, L. 2001. Random Forests. Machine Learning 45(1), 5-32.

Brunk, C., Kelly, J., & Kohavi, R. 1997. MineSet: an integrated system for data mining. In D. Heckerman, H. Mannila, D. Pregibon (Eds.) Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97), AAAI Press, California, 135-138.

Chan, P. & Stolfo, S., 1997. On the accuracy of meta-learning for scalable data mining. Intelligent Information Systems, 8, 5-28.

Clementine User Guide, Version 5. 1998. Integral Solutions Limited.

Cost, S. & Salzberg, S. 1993. A weighted nearest neighbor algorithm for learning with symbolic features. Machine Learning 10(1), 57-78.

CRISP-DM. 2004. Cross Industry Standard Process for Data Mining; see www.crisp-dm.org.

Dasgupta, S. 1999. Learning Mixtures of Gaussians, Proceedings of 40th Annual Symposium on Foundations of Computer Science, 634.

Dasgupta, S. 2000. Experiments with Random Projection, Proceedings of 16th Conference on Uncertainty in Artificial Intelligence, 143-151.

Dasgupta, S. & Gupta, A. 2003. An elementary proof of a theorem of Johnson and Lindenstrauss. Random Structures and Algorithms, 22(1), 60-65.

Deerwester, S., Dumais, S., Furnas, G., Landauer, T. & Harshman, R. 1990. Indexing by latent semantic analysis. Journal of the American Society for Information Science, 41(6), 391-407.

Diamantras, K. & Kung S. 1996. Principal Component Neural Networks. John Wiley & Sons.

Dietterich, T. 1997. Machine learning research: four current directions, AI Magazine 18(4), 97-136.

Dietterich, T. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation, 10 (7), 1895-1923.

Domingos, P. & Pazzani, M. 1996. Beyond independence: conditions for the optimality of the simple Bayesian classifier. In L. Saitta (Ed.) Proceedings of 13th International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann, 105-112.

Domingos, P. & Pazzani, M. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29 (2,3), 103-130.

Duda, R., Hart, P. & Stork, D. 2001. Pattern classification, 2nd edition. Wiley, New York.

Eklund, P. 1999. Comparative study of public domain supervised machine-learning accuracy on the UCI database. In B. Dasarathy (Ed.) Data mining and knowledge discovery: theory, tools, and technology. Proceedings of SPIE, Vol. 3695. Bellingham, WA: SPIE, 39-50.

Fayyad, U. 1996. Data Mining and Knowledge Discovery: Making Sense Out of Data, IEEE Expert 11(5), 20-25.

Fayyad, U., Grinstein G. & Wierser A. 2001. Information Visualization in Data Mining and Knowledge Discovery. San Diego, Morgan Kaufmann.

Fayyad, U. & Uthurusamy, R. 2002. Evolving data into mining solutions for insights. Communications of the ACM 45(8), 28-31.

Fisher, R. 1936. The Use of Multiple Measurements in Taxonomic Problems, Annals of Eugenics, 7(2), 179-188.

Fradkin, D. & Madigan, D. 2003. Experiments with random projections for machine learning. Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press, Washington D.C., 517-522.

Friedman, J. 1997. On bias, variance, 0/1-loss, and the curse of dimensionality. Data Mining and Knowledge Discovery 1 (1), 55-77.

Fukunaga, K. 1990. Introduction to statistical pattern recognition. 2nd Edition, New York, Academic Press.

Gaede, V. & Günther, O. 1998. Multidimensional access methods, ACM Comput. Surv. 30 (2), 170-231.

Gama, J. 1999. Combining classification algorithms. Dept. of Computer Science, University of Porto, Portugal. PhD thesis.

Grossman, R., Hornick, M. & Meyer, G. 2002. Data mining standards initiatives, Communications of the ACM, 45(8), 59-61.

Hall, P. & Li, K. 1993. On Almost Linearity of Low Dimensional Projections From High Dimensional Data, The Annals of Statistics, 21(2), 867-889.

Ho, T. 1998. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(8), 832–844.

Hughes, G. 1968. On the mean accuracy of statistical pattern recognizers. IEEE Transactions on Information Theory, 14(1), 55-63.

Hyvärinen, A., Karhunen, J. & Oja, E. 2001. Independent Component Analysis. New York: John Wiley & Sons, Inc.

Iivari, J., Hirscheim, R. & Klein, H. 1999. A paradigmatic analysis contrasting information systems development approaches and methodologies, Information Systems Research 9(2), 164-193.

Imielinski, T. & Mannila, H. 1996. A database perspective on knowledge discovery. Communications of the ACM, 39(11), 58-64.

Indyk, P. & Motwani, R. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality, Proceedings of the 30th ACM symposium on Theory of Computing, 604-613.

Ives, B., Hamilton, S. & Davis, G. 1980. A Framework for Research in Computer-based Management Information Systems", Management Science 26(9), 910-934.

Jimenez, L. & Landgrebe, D. 1995. High dimensional feature reduction via projection pursuit. School of electrical and computer engineering, Purdue University, TR-ECE 96-5.

John, G. 1997. Enhancements to the data mining process. Dept. of Computer Science, Stanford University, Stanford, USA. PhD Thesis.

Jolliffe, I. 1986. Principal Component Analysis. New York: Springer.

JSR. 2004. Java Specification Request 73; also available from http://jcp.org/en/jsr/detail?id=073.

Kalousis, A. 2002. Algorithm Selection via Meta-Learning. University of Geneve, Department of Computer Science. PhD Thesis.

Kiang, M. 2003. A comparative assessment of classification methods, Decision Support Systems 35, 441-454

Kleinberg, J. 1997. Two algorithms for nearest-neighbor search in high dimensions, Proceedings of the 29th ACM symposium on Theory of Computing, 599-608.

Kleinberg, E. 2000. On the algorithmic implementation of stochastic discrimination. IEEE Transactions on PAMI 22 (5), 473–490

Kohavi, R. 1995a. A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. Mellish (Ed.), Proceedings of 14th International Joint Conference on Artificial Intelligence IJCAI-95, San Francisco, Morgan Kaufmann, 1137-1145.

Kohavi, R. 1995b. Wrappers for performance enhancement and oblivious decision graphs. Dept. of Computer Science, Stanford University, Stanford, USA. PhD Thesis.

Kohavi, R. & John, G. 1998. The wrapper approach. In: H. Liu & H. Motoda (Eds.) Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 33-50.

Kohavi, R., Sommerfield, D. & Dougherty, J. 1996. Data mining using MLC++: a machine learning library in C++. In M. Radle (Ed.) Proceedings of 8th IEEE Conference on Tools with Artificial Intelligence. Los Alamitos: IEEE CS Press, 234-245.

Kononenko, I. 1993. Inductive and Bayesian learning in medical diagnosis. Applied Artificial Intelligence 7(4), 317-337.

Kurimo, M. 1999. Indexing audio documents by using latent semantic analysis and SOM. In: E. Oja & S. Kaski, (Eds.), Kohonen Maps, Elsevier, 363-374.

Liu, H. 1998. Feature Extraction, Construction and Selection: A Data Mining Perspective, Boston, Kluwer Academic Publishers.

Liu, H., Motoda H. & Yu L. 2004. A selective sampling approach to active feature selection, Artificial Intelligence 159(1-2), 49-74.

Melton, J. & Eisenberg, A. 2001. SQL Multimedia and Application Packages (SQL/MM). ACM Sigmod Record, SIGMOD Record 30(4), 97-102.

Merz, C. 1998. Classification and regression by combining models, Dept. of Information and Computer Science, University of California, Irvine, USA, PhD Thesis.

Michalski, R. 1997. Seeking Knowledge in the Deluge of Facts, Fundamenta Informaticae 30, 283-297.

Mitchel, T. 1997. Machine Learning. McGraw-Hill.

Nunamaker, W., Chen, M., & Purdin, T. 1990-91. Systems development in information systems research, Journal of Management Information Systems 7(3), 89-106.

Oja, E. 1997. The nonlinear PCA learning rule in independent component analysis. Neurocomputing, 17(1), 25-46.

OLE DB. 2004. OLE DB for Data Mining Specification 1.0. Microsoft; www.microsoft.com/data/oledb/default.htm.

Opitz, D. & Maclin, D. 1999. Popular ensemble methods: an empirical study. Journal of Artificial Intelligence Research 11, 169-198.

Oza, N. & Tumer, K. 1999. Dimensionality reduction through classifier ensembles. Computational Sciences Division, NASA Ames Research Center, Moffett Field, CA. TR NASA-ARC-IC-1999-124.

Papadimitriou, C., Tamaki, H., Raghavan, P. & Vempala, S. 1998. Latent semantic indexing: a probabilistic analysis, Proceedings of 17th ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems, 159-168.

Pechenizkiy, M. 2005a. Data mining strategy selection via empirical and constructive induction. In: M.H. Hamza (Ed.) Proceedings of the IASTED

International Conference on Databases and Applications DBA'05, ACTA Press, 59-64.

Pechenizkiy, M. 2005b. The Impact of the Feature Extraction on the Performance of a Classifier: kNN, Naïve Bayes and C4.5. In: B. Kegl & G. Lapalme (Eds.): Proceedings of 18th CSCSI Conference on Artificial Intelligence AI'05, LNAI 3501, Heidelberg: Springer-Verlag, 268-279.

Pechenizkiy, M., Puuronen S. & Tsymbal, A. 2003a. Feature extraction for classification in knowledge discovery systems, In: V.Palade, R.Howlett, L.Jain (Eds.), Proceedings of 7th International Conference on Knowledge-Based Intelligent Information and Engineering Systems KES'2003, LNAI, Vol.2773, Heidelberg: Springer-Verlag, 526-532.

Pechenizkiy, M., Puuronen, S. & Tsymbal, A. 2003b. Feature Extraction for Classification in the Data Mining Process. International Journal on Information Theories and Applications 10(1), Sofia, FOI-Commerce, 321-329.

Pechenizkiy, M., Puuronen, S. & Tsymbal, A. 2005a. On the Use of Information Systems Research Methods in Data Mining. In: O.Vasilecas *et al.* (Eds.) Proceedings of 13th International Conference on Information Systems Development: Advances in Theory, Practice and Education ISD'04, Springer, 487-499.

Pechenizkiy, M., Puuronen, S. & Tsymbal, A. 2006. The Impact of Sample Reduction on PCA-based Feature Extraction for Supervised Learning. (to appear) In: H. Haddad *et al.* (Eds.), Proceedings of 21st ACM Symposium on Applied Computing (SAC'06, Data Mining Track), ACM Press.

Pechenizkiy, M., Tsymbal, A. & Puuronen S. 2004. PCA-based feature transformation for classification: issues in medical diagnostics, In: R. Long et al. (Eds.), Proceedings of 17th IEEE Symposium on Computer-Based Medical Systems CBMS'2004, Bethesda, MD, IEEE CS Press, 2004, 535-540.

Pechenizkiy, M., Tsymbal, A. & Puuronen, S. 2005b. Knowledge Management Challenges in Knowledge Discovery Systems. In: IEEE Workshop Proceedings of DEXA'05, 6th Int. Workshop on Theory and Applications of Knowledge Management TAKMA'05, IEEE CS Press, 433-437.

Pechenizkiy M., Tsymbal A., Puuronen S. 2005c. Local Dimensionality Reduction within Natural Clusters for Medical Data Analysis, In: P.Cunningham & A.Tsymbal (Eds.), Proc. 18th IEEE International Symposium on Computer-Based Medical Systems CBMS'2005, Los Alamitos, CA: IEEE CS Press, 365-370.

Pechenizkiy, M., Tsymbal, A. & Puuronen, S. 2005d. On Combining Principal Components with Fisher's Linear Discriminants for Supervised Learning. (submitted to) Special Issue of Foundations of Computing and Decision Sciences "Data Mining and Knowledge Discovery" (as extended version of Pechenizkiy *et al.*, 2005e).

Pechenizkiy, M., Tsymbal, A. & Puuronen, S. 2005e. On Combining Principal Components with Parametric LDA-based Feature Extraction for Supervised Learning. In: T.Morzy *et al.* (Eds.), Proceedings of 1st ADBIS

Workshop on Data Mining and Knowledge Discovery, ADMKD’05, Tallinn, Estonia, 47-56.

Pechenizkiy, M., Tsymbal, A. & Puuronen, S. 2005f. Supervised Learning and Local Dimensionality Reduction within Natural Clusters: Biomedical Data Analysis, (submitted to) IEEE Transactions on Information Technology in Biomedicine, Special Post-conference Issue "Mining Biomedical Data" (as extended version of Pechenizkiy *et al.*, 2005c).

Pechenizkiy, M., Tsymbal, A., Puuronen, S. & Patterson, D. 2005g. Feature Extraction for Dynamic Integration of Classifiers, (submitted to) Fundamenta Informaticae, IOS Press.

Pechenizkiy, M., Tsymbal, A., Puuronen, S., Shifrin, M. & Alexandrova, I. 2005h. Knowledge Discovery from Microbiology Data: Many-sided Analysis of Antibiotic Resistance in Nosocomial Infections. In: K. Althoff *et al.* (Eds.) Post-Conference Proceedings of 3rd Conference on Professional Knowledge Management: Experiences and Visions, LNAI 3782, Heidelberg: Springer-Verlag, 360-372.

Piatetsky-Shapiro, G. 2000. Knowledge Discovery in Databases: 10 years after. SIGKDD Explorations 1(2), 59-61.

PMML. 2004. Predictive Model Markup Language. Data Mining Group, see www.dmg.org.

Popelinsky, L. 2001. Combining the Principal Components Method with Different Learning Algorithms. In Proceedings of 12th European Conference on Machine Learning, Workshop on Integrating Aspects of Data Mining, Decision Support and Meta-Learning,.

Puuronen, S., Terziyan, V. & Tsymbal, A. 1999. A dynamic integration algorithm for an ensemble of classifiers. In Z.W.Ras & A.Skowron (Eds.) Foundations of intelligent systems: 11th International Symposium ISMIS’99, Warsaw, Poland. LNAI 1609. Berlin: Springer, 592-600.

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

Quinlan, J. 1996. Bagging, boosting, and C4.5. In Lenz (Ed.) Proceedings of the 13th National Conference on Artificial Intelligence, AAAI-96, New York, NY: Springer-Verlag, AAAI Press, 725-730.

Salzberg, S. 1999. On comparing classifiers: a critique of current research and methods. Data Mining and Knowledge Discovery 1, 1-12.

Schaffer, C. 1993. Selecting a classification method by cross-validation, Machine Learning 13, 135-143.

Skurichina, M. & Duin, R. 2001. Bagging and the random subspace method for redundant feature spaces, in: J. Kittler, F. Roli (Eds.), Proceedings of 2nd International Workshop on Multiple Classifier Systems MCS’01, Cambridge, UK, 1–10.

Thrun, S., Bala, J, Bloedorn, E., *et al.* 1991. The MONK’s problems – a performance comparison of different learning algorithms. Carnegie Mellon University, Pittsburg PA. Technical report CS-CMU-91-197.

Tkach, D. 1998. Information Mining with the IBM Intelligent Miner Family. An IBM Software Solutions White Paper.

Tsymbal, A. 2002. Dynamic Integration of Data Mining Methods in Knowledge Discovery Systems, Jyväskylä, University of Jyväskylä. PhD Thesis.

Tsymbal, A., Puuronen, S. & Skrypnyk, I. 2001. Ensemble feature selection with dynamic integration of classifiers. In International ICSC Congress on Computational Intelligence Methods and Applications, 558-564.

Tsymbal, A., Pechenizkiy, M., Puuronen, S. & Patterson, D. 2003. Dynamic integration of classifiers in the space of principal components, In: L.Kalinichenko, *et al.* (Eds.), Proceedings of Advances in Databases and Information Systems: 7th East-European Conference ADBIS'03, LNCS, Vol. 2798, Heidelberg: Springer-Verlag, 278-292.

Tsymbal, A., Puuronen, S., Pechenizkiy, M., Baumgarten, M. & Patterson, D. 2002. Eigenvector-based Feature Extraction for Classification. In: S.M. Haller, G. Simmons (Eds.), Proceedings of 15th International FLAIRS Conference on Artificial Intelligence, AAAI Press, 354-358.

Turney, P. 1996. The management of context-sensitive features: A review of strategies. In: Proceedings of Workshop on Learning in Context-Sensitive Domains at the 13th International Conference on Machine Learning, 60-66.

Vijayakumar, S. & Schaal, S. 1997. Local Dimensionality Reduction for Locally Weighted Learning, Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, 220-225.

Weingessel, A. & Hornik, K. 1998. Local PCA Algorithms. IEEE Transactions on Neural Networks 8(5), 1208-1211.

WEKA 3. 2004. Data Mining Software in Java. Also available from http://www.cs.waikato.ac.nz/ml/weka/

Witten, I. & Frank, E. 2000. Data Mining: Practical machine learning tools with Java implementations, San Francisco: Morgan Kaufmann.

Wolpert, D. & MacReady, W. 1996. No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1), 67-82.

Zorman, M., Eich, H., Kokol, P. & Ohmann, C. 2001. Comparison of Three Databases with a Decision Tree Approach in the Medical Field of Acute Appendicitis, In: Patel *et al.* (Eds.), Proceedings of 10th World Congress on Health and Medical Informatics, Vol.2, Amsterdam: IOS Press, 1414-1418.

APPENDIX A. DATASETS USED IN THE EXPERIMENTS

The majority of the datasets used in the experiments were taken from the University of California at Irvine Machine Learning Repository (Blake & Merz, 1998). The *Acute Abdominal Pain* (AAP) datasets were provided by Laboratory for System Design, Faculty of Electrical Engineering and Computer Science, University of Maribor, Slovenia and Theoretical Surgery Unit, Dept. of General and Trauma Surgery, Heinrich-Heine University Düsseldorf, Germany (Zorman *et al.*, 2001). The *Antibiotic Resistance* dataset was collected in the Hospital of N.N. Burdenko Institute of Neurosurgery, Moscow, Russia. The main characteristics of the datasets used in experiments throughout the thesis are presented in Table 1.

The table includes the name of the dataset, the number of instances included in the dataset, the number of different classes of instances, and the numbers of different kinds of features included in the instances.

The Acute Abdominal Pain (AAP) datasets represent the same problem of separating acute appendicitis (class "appendicitis"), which is a special problem of acute abdominal pain, from other diseases that cause acute abdominal pain (class "other diagnoses"). The early and accurate diagnosis of acute appendicitis is still a difficult and challenging problem in everyday clinical routine. AAPI, AAPII and AAPIII are three large data sets with cases of acute abdominal pain (AAP): (1) Small-AAP I; (2) Medium-AAP II; and (3) Large-AAP III, with the numbers of instances respectively 1254, 2286, and 4020 (Zorman *et al.*, 2001). These data sets represent the same problem of separating acute appendicitis from other diseases that cause acute abdominal pain. The data for AAP I has been collected from 6 surgical departments in Germany, for AAP II – from 14 centers in Germany, and for AAP III – from 16 centers in Central and Eastern Europe. Each data set includes 18 features from history-taking and clinical examination (Zorman *et al.*, 2001). These features are standardized by the World Organization of Gastroenterology (OMGE).

The Antibiotic Resistance dataset was collected in the Hospital of N.N. Burdenko Institute of Neurosurgery, Moscow, Russia, using the analyzer Vitek-60 (developed by *bioMérieux*, www.biomerieux.com) over the years 1997-2003 and the information systems Microbiologist (developed by the Medical Informatics Lab of the institute) and Microbe (developed by the Russian company MedProject-3). Each instance of the data used in analysis represents one sensitivity test and contains the following features: pathogen that is isolated during the bacterium identification analysis, antibiotic that is used in the sensitivity test and the result of the sensitivity test itself (sensitive S, resistant R or intermediate I), obtained from Vitek according to the guidelines of the National Committee for Clinical Laboratory Standards (NCCLS). Information about sensitivity analysis is connected with patient, his or her demographical data (sex, age) and hospitalization in the Institute (main department, days spent in ICU, days spent in the hospital before test, etc.) Each bacterium in a sensitivity test in the database is isolated from a single specimen that may be blood, liquor, urine, etc. In this study we focus on the analysis of meningitis cases only, and the specimen is liquor. For this purposes we picked up 4430 instances of sensitivity tests related to the meningitis cases of the period January 2002 – August 2004. We introduced 5 grouping binary features for pathogens and 15 binary features for antibiotics. These binary features represent hierarchical grouping of pathogens and antibiotics into 5 and 15 categories respectively. Thus, each instance in the dataset had 34 features that included information corresponding to a single sensitivity test augmented with the data concerning the used antibiotic, the isolated pathogen, the sensitivity test result and clinical features of the patient and his/her demographics.

TABLE 1 Basic characteristics of the datasets

Dataset	Instances	Classes	Features			
			Categorical	Numerical	Total	Total*
Acute Abdominal Pain I	1251	2	17	1	18	89
Acute Abdominal Pain II	2279	2	17	1	18	89
Acute Abdominal Pain III	4020	2	17	1	18	89
Antibiotic Resistance	4430	3	28	6	34	47
Balance	625	3	0	4	4	3
Breast Cancer Ljubljana	286	2	9	0	9	38
Car Evaluation	1728	4	6	0	6	21
Pima Indians Diabetes	768	2	0	8	8	8
Glass Recognition	214	6	0	9	9	9
Heart Disease	270	2	8	5	5	13
Ionosphere	351	2	0	34	34	33
Iris Plants	150	3	0	4	4	4
Kr-vs-kp	3196	2	36	0	36	38
LED	300	10	7	0	7	7
LED17	300	10	24	0	24	24
Liver Disorders	345	2	0	6	6	6
Lymphography	148	4	15	3	18	36
MONK-1	432	2	6	0	0	15
MONK-2	432	2	6	0	0	15
MONK-3	432	2	6	0	0	15
Soybean	47	4	0	35	35	35
Thyroid	215	3	0	5	5	5
Tic-Tac-Toe Endgame	958	2	9	0	9	27
Vehicle	846	4	0	18	18	18
Voting	435	2	16	0	16	17
Zoo	101	7	16	0	16	17

* when categorical features are binarized.

The Balance dataset was generated to model psychological experimental results. Each example is classified as having the balance scale tip to the right, tip to the left, or be balanced. The attributes are the left weight, the left distance, the right weight, and the right distance. The correct way to find the class is the greater of (left-distance * left-weight) and (right-distance * right-weight). If they are equal, it is balanced.

In the Breast Cancer Ljubljana dataset the task is to determine whether breast cancer will or will not recur. The data were originally obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia.

The Car Evaluation dataset was derived from a simple hierarchical decision model that evaluates cars according to a concept structure. The Car Evaluation dataset contains examples with the structural information removed, i.e., directly relates a car to the six input attributes: buying, maint, doors, persons, lug_boot, and safety. The four classes are "unacceptable", "acceptable", "good", and "very good".

The task for the Pima Indians Diabetes dataset is to determine whether the patient shows signs of diabetes according to World Health Organization criteria. There are eight continuous features: number of times pregnant, plasma glucose concentration, diastolic blood pressure, triceps skin fold thickness, 2-hour serum insulin, body mass index, diabetes pedigree function, and age.

The DNA dataset is drawn from the field of molecular biology. Splice junctions are points on a DNA sequence at which "superfluous" DNA is removed during protein creation. The task is to recognize exon/intron boundaries, referred to as EI sites; intron/exon boundaries, referred to as IE sites; or neither. The features provide a window of 60 nucleotides. The classification is the middle point of the window, thus providing 30 nucleotides at each side of the junction.

In the Glass Recognition dataset the task is to identify which one of the six types of glass is present from the chemical elements in a sample.

The task for the Heart Disease dataset is to distinguish the presence or absence of heart disease in patients. The features include: age, sex, chest pain type, resting blood pressure, fasting blood sugar, max heart rate, *etc*.

The Ionosphere dataset includes radar data that was collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 high-frequency antennas with a total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing evidence of some type of structure in the ionosphere. "Bad" returns are those that do not; their signals pass through the ionosphere. Received signals were processed using an autocorrelation function whose arguments are the time of a pulse and the pulse number. There were 17 pulse numbers for the Goose Bay system. Instances in this dataset are described by 2 attributes per pulse number, corresponding to the complex values returned by the function resulting from the complex electromagnetic signal.

The Iris Plants dataset created by R.A. Fisher is perhaps the best known database in the machine learning literature. The task is to classify iris plants into one of three iris plants varieties: Iris Setosa, Iris Versicolour, and Iris Virginica. This is an exceedingly simple domain and very low error rates have been reached already long ago.

The Kr-vs-kp dataset represents one classical type of chess end-game – King with Rook versus King with Pawn on a7 and usually abbreviated as KRKPA7. The pawn on a7 means it is one square away from becoming a queen. It is the King with Rook's side (white) to move. White is deemed to be unable to win if the Black pawn can safely advance.

The LED dataset contains data about the LED display problem, where the goal is to learn to recognize decimal digits having information about whether the seven corresponding LED segments are on or off. The LED 17 dataset represents an extension of the LED display problem, with an additional 17 irrelevant attributes being added to the instance space. These attributes are randomly assigned the values of 0 or 1.

The Liver Disorders dataset was created by BUPA Medical Research Ltd, and the task is to predict liver disorders that might arise from excessive alcohol consumption.

The Lymphography dataset was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. There are 15 categorical and 3 numerical attributes, and the classes being predicted are: "normal find", "metastases", "malign lymph", and "fibrosis".

The MONK's problems are a collection of three artificial binary classification problems over the same six-attribute discrete domain (a1,…,a6). All MONK's datasets contain 432 instances without missing values, representing the full truth tables in the space of the attributes. The "true" concepts MONK-1, MONK-2, and MONK-3 underlying each MONK's problem are given by: (a1=a2)or(a5=1) for MONK-1, exactly two of {a1=1, a2=1, a3=1, a4=1, a5=1, a6=1} for MONK-2, and (a5=3 and a4=1)or(a5<>4 and a2<>3) for MONK-3. MONK-3 has 5% additional noise (misclassifications) in the training set. The MONK's problems were the basis of the first international comparison of learning algorithms (Thrun *et al.*, 1991).

The Soybean dataset includes data about the soybean disease diagnosis. This is a small subset of the original Soybean-large database. There are 35 numerical attributes, and 4 classes, representing soybean diseases.

In the Thyroid dataset, five laboratory tests are used to try to predict whether a patient's thyroid is in the class "euthyroidism", "hypothyroidism" or "hyperthyroidism". The diagnosis (the class label) is based on a complete medical record, including anamnesis, scan etc.

The Tic-Tac-Toe Endgame dataset encodes the complete set of possible board configurations at the end of tic-tac-toe games, where "x" is assumed to have played first. The target concept is "win for x" (i.e., true when "x" has one of 8 possible ways to create a "three-in-a-row"). The dataset contains 958 instances without missing values, each with 9 attributes, corresponding to tic-tac-toe squares and taking on 1 of 3 possible values: "x", "o", and "empty".

In the Vehicle dataset, the goal is to classify a given silhouette as one of four types of vehicles ("Opel", "Saab", "Bus", and "Van"), using a set of 18 numerical features extracted from the silhouette. The vehicle may be viewed from one of many different angles. This dataset comes from the Turing Institute, Glasgow, Scotland.

The Voting dataset includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac in 1984. The goal is build a classification model to predict the voting congressman to be either a democrat or a republican.

Zoo is a simple dataset created by Richard S. Forsyth with instances containing 17 Boolean valued-attributes, and representing 7 types of animals.

A survey of widely used learning algorithms (decision trees, neural networks, and rule-based classifiers) on twenty-nine datasets from the UCI machine learning repository is given in Eklund (1999). This survey connects the properties of datasets examined with the selection of learning algorithms. In (Salzberg, 1999) the use of the datasets from the UCI repository was strongly criticized, because in his opinion it is difficult to produce major new results using well-studied and widely shared data. Hence, a new "significant" finding may occur to be "a statistical accident". However, we would prefer to interpret this message as a caution to be careful when stating final conclusions if research has been conducted only on such benchmarks.

YHTEENVETO (FINNISH SUMMARY)

Tiedon louhinnalla pyritään paljastamaan tietokannasta tietomassaan sisältyviä säännönmukaisuuksia joiden olemassaolosta ei vielä olla tietoisia. Mikäli tietokantaan sisältyvät tiedot ovat kovin moniulotteisia sisältäen lukuisia piirteitä, heikkenee monien koneoppimisen menetelmien suoriutumiskyky ratkaisevasti. Ilmiötä nimitetään ″moniulotteisuuden kiroukseksi″ (″curse of dimensionality″) sen johtaessa usein sekä laskennallisen kompleksisuuden että luokitusvirheiden kasvuun. Toisaalta tietokantaan mahdollisesti sisältyvät epärelevantit tai vain epäsuorasti relevantit piirteet tarjoavat heikon esitysavaruuden tietokannan käsiterakenteen kuvaamiseen.

Tutkimuksen tavoitteena on kehittää tietoainesta kuvaavien piirteiden muodostamisen teoreettista taustaa ja käytännön toteutusta. Tällä piirteiden muodostamisella pyritään joko ulotteisuuden pienentämiseen tai esitysavaruuden parantamiseen (tai molempiin) ohjatun koneoppimisen tarpeita varten. Työssä sovelletaan perinteistä pääkompponenttianalyysiä ja kahta luokkiin liittyvää tietoa hyödyntävää analyysimenetelmää. Tarkastelu ulotetaan sekä perustason luokittelijan muodostamiseen että luokittelijakokoelmaan siihen sisältyvien luokittelijoiden integroinnin näkökulmasta. Tarkastelun teoreettisen perustan muodostavat tiedon louhinnan, koneoppimisen ja hahmontunnistuksen tutkimusalueet. Tutkimuksen yhteydessä kokeellista osuutta varten laadittu ohjelmisto on rakennettu Javalla toteutetulle avoimen koodin koneoppimisohjelmistoalustalle (WEKA).

Työ koostuu erillisistä artikkeleista ja niihin tukeutuvasta yhteenvedosta, jossa tutkimuksen tulokset kootaan asetettujen tutkimusongelmien (TOi) alle. Tutkimusongelmista viisi ensimmäistä on enemmän teoriapainotteista ja viisi seuraavaa enemmän käytäntöpainotteisia. Tässä suomenkielisessä yhteenvedossa esitetään sekä tutkimusongelma että työn tulos ongelma ongelmalta (tekstissä tietokanta tarkoittaa saman piirrerakenteen omaavaa tapausten joukkoa):

TO1: Kuinka tärkeää on luokkainformaation käyttö piirteiden muodostamisprosessissa? Tutkimuksessa todettiin luokkainformaation olevan ratkaisevan tärkeää useiden tietokantojen tapauksessa. Luokkiin liittyvää tietoa hyödyntävien piirteiden muodostamisprosessin todettiin voivan johtaa tarkempaan luokittelijaan niille tietokannoille, joille puhtaasti varianssiin perustuva piirteiden muodostamisprosessi ei vaikuta luokittelijan tarkkuuteen tai heikentää sitä.

TO2: Onko piirteiden muodostaminen tieto- vai hypoteesivetoista konstruktiivista induktiota? Tutkimuksessa todettiin, että piirteiden muodostamisprosessin tulisi sopia niin tietokantaan kuin ohjatun koneoppimisen menetelmäänkin, koska piirteiden muodostamisprosessien keskinäinen paremmuusjärjestys vaihteli paljon molempien suhteen.

TO3: Onko piirteiden muodostaminen hyödyllistä luokittelijakokoelman dynaamisen integroinnin yhteydessä niin kuin se on yksittäisten luokittelijoiden tapauksessa? Todettiin, että piirteiden muodostaminen voi johtaa tarkkuuden parantumiseen dynaamisen integroinnin yhteydessä niille tietokannoille, joille piirteiden muodostaminen johtaa tarkkuuden parantumiseen yksittäisen tapauspohjaisen luokittelijan yhteydessä.

TO4: Ovatko alkuperäiset piirteet, muodostetut piirteet vai kummatkin käyttökelpoisia ohjatun koneoppimisen yhteydessä? Alkuperäisten ja muodostettujen piirteiden kombinaatio voi olla hyväksi ohjatun koneoppimisen yhteydessä joillekin tietokannoille. Erityisesti käyttökelpoisuus todettiin päätöspuutyyppisten luokittelijoiden, kuten C4.5 käytön yhteydessä. Muutaman pääkomponentin yhdistämisen lineaaristen erottimien tueksi todettiin voivan johtaa parempaan luokitustarkkuuteen C4.5-tyyppisen luokittelijan yhteydessä. Samalla kuitenkin todettiin ettei yhdistäminen kannata Bayes-tyyppisen luokittelijan yhteydessä koska se johtaa tässä yhteydessä luokittelijan epästabiiliin käyttäytymiseen.

TO5: Kuinka moni muodostettu piirre on käyttökelpoinen koneoppimisen yhteydessä? Piirteiden muodostamisprosessin yhteydessä käytettävät kynnysarvot piirteiden selittämän varianssin osalta vaihtelivat paljon tietokannasta toiseen.

TO6: Kuinka selviytyä silloin kun tietoon sisältyy kontekstia kuvaavia piirteitä ja ongelma-avaruuden heterogeenisuutta? Luonteva ryvästely osoittautui hyvin tehokkaaksi lähestymistavaksi laadittaessa paikallisia ratkaisuja piirteiden muodostamissa ja ohjatussa koneoppimisessa.

TO7: Miten otoksen pienentäminen vaikuttaa piirteiden muodostamiseen ohjattua koneoppimista varten? Yleensä epäparametrinen piirteiden muodostaminen johtaa samaan tai parempaan tarkkuuteen pienemmällä oppimistapausten lukumäärällä kuin parametrinen piirteiden muodostaminen. Tulokset osoittivat, että silloin kun piirteiden muodostamiseen ja luokittelijan rakentamiseen käytettävien oppimistapausten osuus on suhteellisen pieni on tärkeää käyttää sopivaa otoksen pienentämistekniikkaa, jotta edustavimmat tapaukset tulevat valituiksi piirteiden muodostamisprosessiin.

TO8: Milloin piirteiden muodostaminen ohjattua koneoppimista varten on käytännöllistä? Alustavien tulosten perusteella näyttäisi olevan vaikea ennustaa milloin piirteiden muodostaminen voi olla käyttökelpoista ohjatun koneoppimisen yhteydessä. Edelleen on vaikea ennustaa mitkä muodostetuista piirteistä ja kuinka monta niistä tulisi ottaa käyttöön ohjatun koneoppimisen yhteydessä.

TO9: Miten tulkita muodostettuja piirteitä? Tutkimuksen perusteella näyttäisi siltä, että käsiteltävästä tietokannasta, alkuperäisten piirteiden merkityksestä ja tarkasteltavasta ongelmasta sekä ohjatun koneoppimisen tekniikasta riippuen piirteiden muodostaminen voi olla tulkinnan kannalta hyödyllistä tai haitallista.

TO10: Miten valita piirteiden muodostamistekniikan ja ohjatun koneoppimistekniikan yhdistelmä käsillä olevaan ongelmaan? Työssä esitetään yleinen viitekehys sopivimman tiedon louhintastrategian valitsemiseksi keräämällä tiedonlouhintatekniikoiden ja niiden kombinaatioiden käyttökokemuksia erilaisten tietokantojen parissa.