JYVÄSKYLÄ STUDIES IN COMPUTING
152

Iryna Skrypnyk

Unstable Feature Relevance in Classification Tasks

JYVÄSKYLÄN YLIOPISTO

JYVÄSKYLÄ STUDIES IN COMPUTING 152

Iryna Skrypnyk

Unstable Feature Relevance in Classification Tasks

Esitetään Jyväskylän yliopiston informaatioteknologian tiedekunnan suostumuksella
julkisesti tarkastettavaksi yliopiston Agora-rakennuksen salissa AgAud 3
joulukuun 21. päivänä 2011 kello 12.

Academic dissertation to be publicly discussed, by permission of
the Faculty of Information Technology of the University of Jyväskylä,
in building Agora, auditorium AgAud 3, on December 21, 2011 at 12 o'clock noon.

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2011

Unstable Feature Relevance in Classification Tasks

JYVÄSKYLÄ STUDIES IN COMPUTING 152

Iryna Skrypnyk

Unstable Feature Relevance in Classification Tasks

UNIVERSITY OF JYVÄSKYLÄ

JYVÄSKYLÄ 2011

Editors
Seppo Puuronen
Department of Computer Science and Information Systems, University of Jyväskylä
Pekka Olsbo, Ville Korkiakangas
Publishing Unit, University Library of Jyväskylä

URN:ISBN:978-951-39-4608-1
ISBN 978-951-39-4608-1 (PDF)

ISBN 978-951-39-4607-4 (nid.)
ISSN 1456-5390

Jyväskylä University Printing House, Jyväskylä 2011

ABSTRACT

Skrypnyk, Iryna
Unstable Feature Relevance in Classification Tasks
Jyväskylä: University of Jyväskylä, 2011, 232 p.
(Jyväskylä Studies in Computing,
ISSN 1456-5390;152)
ISBN 978-951-39-4607-4 (nid.)
ISBN 978-951-39-4608-1 (PDF)
Finnish summary
Diss.

Over the last decade, data mining has gone through a significant transformation influenced by advanced data collection technologies. Today data mining faces the challenge of dealing with increasingly complex data structures. As a result, data often exhibits instability in measured attribute values (features). In other words, the set of relevant features is not the same through the entire set of domain examples. Considering this problem from another angle, data includes regions with local properties that, in particular, differ from each other with regard to the feature relevance profiles. Global models, therefore, cannot reflect the essential knowledge about the data structure. This thesis presents a description of the unstable feature relevance problem in classification tasks, elaborating the concept of heterogeneous classification problems and introducing different types of feature space heterogeneity. It also suggests a multi-model solution derived from the definition of a subproblem as a group of instances with easier class discrimination and lower complexity in the subspace of locally relevant features. The solution is presented within an ensemble learning framework. The search strategies, suggested for decomposition of classification problems with unstable feature relevance, express different levels of granularity with respect to classes. Evaluation of the candidate subproblems is executed through profiles of feature relevance. These profiles are vectors of weights obtained from feature merit measures and, alternatively, a result of distance metric adaptation. Additional measures of complexity, including class boundaries and density-based measures, are suggested to evaluate decomposition and to serve as preliminary heterogeneity tests. This research contributes towards reaching complementary data analysis goals on classification problems and revealing important insights on the data structure and its complexity. The effects on classification performance were studied through numerous experiments on synthetic, benchmark, and real data from a biomedical research domain. It was found that extraction of subproblems is possible in many cases and it provides meaningful data partitioning results. In many cases it also leads to improvement in predictive performance.

Keywords: feature relevance, feature selection, feature weighting, classification, clustering, ensemble learning, data mining, knowledge discovery, machine learning

Author's address Skrypnyk, Iryna, Ph.Lic.
Department of Computer Science and
Information Systems
University of Jyväskylä, Finland
P.O. Box 35 (Agora), 40014 University of Jyväskylä
iryna.v.skrypnyk@jyu.fi

Supervisors Prof. Puuronen, Seppo, Ph.D.
Department of Computer Science and
Information Systems
University of Jyväskylä, Finland

Prof. Neittanmäki, Pekka, Ph. D.
Department of Mathematical Information Technology
University of Jyväskylä, Finland

Reviewers Prof. Tatiana A. Gavrilova, Ph.D.
Department of Information Technologies in Management
St. Petersburg University, Russia

Prof. Neta Rabin, Ph.D.
Department of Mathematics
Yale University, CT, USA

Opponent Prof. Ruotsalainen, Keijo, Ph.D.
Department of Electrical and Computer Engineering
University of Oulu, Finland

ACKNOWLEDGEMENTS

There are many people to whom I would like to express my sincere gratitude for inspiring, advising, encouraging, and supporting me throughout my work on this dissertation. I thank Prof. Seppo Puuronen for introducing me to the world of research many years ago, for teaching and supervising, his tolerance, gentle guidance, and long hours of fruitful data mining discussions. I would like to thank Prof. Pekka Neittanmäki, who helped bring it to its final form, being an incredibly resourceful and demanding advisor. Thanks to his commitment and support, I continued working on my project with inspiration.

I would like to thank Prof. Amir Averbuch of Tel Aviv University, Prof. Gil David, Prof. Tommi Kärkkäinen, and Dr. Sami Äyrämö of the University of Jyväskylä for their valuable feedback on my work. I thank my reviewers, Prof. Neta Rabin of Yale University and Prof. Tatiana Gavrilova of St. Petersburg University, for their time and positive criticism.

There were two especially influential research collaborations I had during my internships and research visits. One was with Dr. Se June Hong, currently a research staff member Emeritus at IBM T.J. Watson Research Center, whom I thank for inviting me for an IBM internship when I first expressed interest in heterogeneity in classification problems. He helped me gain professional insight, while also sharing his wisdom about life. I also thank Dr. Hong for being a reviewer of my Licentiate thesis, which served as a first step toward, and the basis of my doctoral research.

Dr. Tin Kam Ho, currently Head of the Statistics and Learning Research Department at Bell Laboratories, Alcatel-Lucent, also significantly influenced my work. She introduced me to several particularly interesting research topics, while hosting my visit at Bell Laboratories twice. She has also become my role model in research and at the personal level.

I would like to acknowledge the generous sponsors who made this research possible: the COMAS graduate school of the University of Jyväskylä, the Ellen and Artturi Nyyssösen Foundation, the IBM T.J. Watson Research Center, NY, and the Center for Discrete Mathematics and Theoretical Computer Science (DIMACS) of Rutgers University, NJ.

I would like to thank all my colleagues and co-authors for productive collaboration. The administrative and technical staff of the University of Jyväskylä has always been particularly friendly, efficient, and prompt. I enjoyed my time at the University of Jyväskylä very much.

Finally, I would like to thank my family and my dear friends for their sincere desire that this work be successful, and for all their support and love. Foremost among these is my husband, Andrei. I do not have the words to express my deep gratitude to him.

Jyväskylä, December 8, 2011.
Iryna Skrypnyk

FIGURES

TABLES

CONTENTS

1 INTRODUCTION

Over the last decade, data mining and knowledge discovery went through an enormous transformation influenced by rapid growth of web/e-commerce, tremendous progress in biology, and an increased power of collecting, storing and analyzing data in general (Piatetsky-Shapiro, 2007). With the advent of high-throughput experimental technologies and of high-speed Internet connections, generation and transmission of large volumes of data have been automated. As a result, science, industry, and even individuals have to face the challenge of dealing with large data sets, which are not only impractical for manual analysis, but also challenging for some automated analysis techniques (Kriegel *et al.*, 2007). Since the first definition of the knowledge discovery process (Fayyad *et al.*, 1996), the concept of a "golden nugget" has evolved, and knowledge has to be extracted now from increasingly complex data.

Modern automated methods for measurement, collection, and analysis of data in all fields of science, industry, and economy are providing more and more data with drastically increasing complexity of structure. This growing complexity is justified on one hand by the need for a richer and more precise description of real-world objects, and on the other hand by the rapid progress in measurement and analysis techniques allowing versatile exploration of objects. (Kriegel *et al.*, 2007)

Intrinsic complexity of a problem may result from an insignificant amount of data, too much data, or ambiguity due to classification problems. In addition, complexity of data may be increased by different factors, including the combination of different data types, accumulation of data from different sources, data having been collected over different periods of time, integration of data in heterogeneous databases, and the pre-processing for further analysis

that sometimes may entail loss or redundancy in information. For example, medical data may include biological analyzes, textual data coming from clinical reports, and image data such as radiographies, echograms, or electrocardiograms. Each type of information needs pre-processing in order to consider these different data simultaneously, thereby encompassing all their complexity.

In order to ensure that measurements in data carry complete information with respect to the entity or phenomenon being analyzed in the problem domain, data is often collected with redundancy. Recent trends of data collection are based on the paradigm "gather whatever possible data, whenever you can". The expectations are that gathered data will have value either for the purpose collected or for a purpose not envisioned. As a result, dimensionality of the data becomes high, while the number of representative examples needed for a consistent problem description and acceptable predictive accuracy level rises exponentially with the number of dimensions (Blayo *et al*., 1995).

In a variety of application domains, data mining deals with data sets having unstable feature relevance across the set of instances with respect to class discrimination (Apte *et al*., 1998; Lazarevič & Obradovič, 2001a; Lazarevič & Obradovič, 2001b). This problem has been recognized by data mining, machine learning and pattern recondition communities for over a decade gaining new meaning nowadays. Many large-scale data analysis problems involve an investigation of relationships between attributes in heterogeneous databases. Large data sets very often exhibit attribute instability, such that the set of relevant attributes is not the same through the entire data space.

For example, in spatial databases different spatial regions may have completely different characteristics. In medical diagnostics data, relevance of attributes depends on context. In heterogeneous databases success of data integration critically depends on the availability of accurate semantic information on data contents (Kim & Seo, 1991). Integration often leads to unstable feature relevance. Problem domains, where predictive models are constructed from heterogeneous data, include bioinformatics, for example, gene functional classification (Pavlidis *et al*., 2001) and prediction of proteins interaction (Thierry-Mieg, 2000). An example from biomedical research is classification of human cancer types using microarray gene expressions (Golub *et al*., 1999; Ramaswamy *et al*., 2001). The number of such domains has lately increased along with the new trends in data collection. Therefore, among the most important characteristics of contemporary real data is heterogeneity due to the data nature and/or source.

Heterogeneous data encompass complexity for modeling with a unimodal approach (Ho *et al*., 2006; Ho & Basu, 2002). There is an ultimate need for improved data analysis techniques which will effectively process heterogeneous data reducing complexity of the analysis problems.

In data mining, a predictive model is constructed using a predefined set of pattern representations (decision rules and trees, similarity-based and probability-based models, and so on). The model is evaluated upon its ability to

discern patterns in the data being analyzed. Classification tasks in data mining are predictive tasks, where the target variable to be predicted takes discrete (categorical) values - classes. Other important analysis tasks, which often accompany a classification task, are finding important dependencies between features representing data, discovering meaningful patterns in data, and acquisition of knowledge about the problem domain.

The approach explored in this thesis aims to discover new patterns in data via decomposition of a complex classification problem onto a number of simpler subproblems, where subproblems themselves serve as patterns or can be viewed as forms of domain knowledge. In particular, classification complexity of labeled data in supervised learning calls for seeking data structure beyond class labels via decomposition of classification problems, that is data partitioning.

This thesis addresses the problem of constructing effective predictive models for heterogeneous data used for classification tasks, disregarding the source of heterogeneity and prior knowledge regarding heterogeneity. In practice, prior knowledge about the problem domain is fairly limited after data passes through a number of pre-processing steps, for example, at the integration stage in multiple heterogeneous database systems.

This chapter introduces the research work performed within the scope of the thesis. Section 1.1 presents the motivation and considers efforts of other researchers in this area. The research questions raised in this thesis are outlined. In Section 1.2 the thesis statement is provided, describing research goal, approach and methods, and a brief overview of the outcomes and contribution. The thesis overview is presented in Section 1.3. Contributions made by the author are summarized in Section 1.4. An overview of author's published works is provided in Section 1.5.

1.1 Motivation

Analysis of literature on machine learning, data mining, and knowledge discovery from databases with respect to recent trends of data collection, new application domains, and new developments enlightened various aspects of the unstable feature relevance problem. It has motivated the study of classification heterogeneity phenomenon in general following with development of a theoretical background, solutions, and details of their practical application. A few most important aspects are highlighted below.

The flexibility of machine learning techniques makes them well suited to applications where little is known *a priori* about the domain, and/or relevant knowledge is hard to elicit (Domingos, 2002). The most important machine learning elaboration for heterogeneous classification problems is that computational power is often better when used to induce multiple models and combine them, instead of adapting a single model (Kuncheva, 2004). Therefore,

decomposition of a classification problem into subproblems for heterogeneous data has been accomplished using an ensemble learning framework.

Currently, the theory of ensemble learning is being explored enthusiastically. As part of this, hierarchical ensembles are being utilized, allowing for improved classification performance as well as the extraction of valuable domain knowledge about relationships and hierarchies among classes, and feature relevance profiles for existing and encoded class combinations (Ghosh, 2002).

Some aspects of the classification heterogeneity problem in contemporary data have been pointed out in earlier machine learning works on local learning and context-sensitive learning (Hastie & Tibshirani, 1996; Friedman, 1994; Turney, 1993). At about the same time the problem was named "classification heterogeneity" and its variations, feature space heterogeneity and class heterogeneity, have been introduced (Apte *et al*., 1998). These notions were kept for elaboration of the heterogeneity concept in the thesis. With the growth and expansion of multiple heterogeneous database systems and large scale data analysis problems, a vision of complex classification problems becomes an interpretation of classification heterogeneity.

In several later works from the data mining and databases research community, different perspectives of the heterogeneity problem have been considered. Commonly, the solutions utilize domain knowledge and require human expertise. Therefore, their application is often restricted to a certain problem domain. For example, in Pavlidis *et al*. (2001) gene functional classification is based on different types of genomic data (yeast phylogenetic profiles and DNA microarray expression) which is analyzed after its decomposition on subsets of domain examples using domain knowledge.

Performance of traditional ensemble learning techniques from different categories of ensemble generation has been investigated in heterogeneous classification problems, such as in Lazarevič *et al*. (2000), and it has been shown that ensemble generated manipulating features (attributes) are potentially advantageous when feature relevance is unstable across the set of domain examples. However, the proposed elaborations on the ensemble techniques manipulating features (Opitz, 1999; O'Sullivan *et al*., 2000) build global models disregarding grouping of instances (domain examples) at homogeneous regions. Another approach (Lazarevič & Obradovič, 2001a) is based on constructing local models, each responsible for a particular region of a heterogeneous data set. Applicability of this approach depends on the success in discovering or approximating those homogeneous regions and their coverage by the local predictive models.

The ensemble technique combining local feature selection and class encoding for class heterogeneity developed in this thesis follows the approach to construct local predictive models for homogeneous regions. A major motivation for this approach is that the subproblems are typically much easier to solve and interpret. Feature weighting, selection, or extraction, can be performed individually for each subproblem as a step of local model

construction. Decomposition of heterogeneous classification problems into subproblems, constructing local models for each subproblem, and using these models for prediction, determine the focus of this research.

This thesis covers the following research issues: variations of classification heterogeneity, data characteristics for evaluating structure of heterogeneous data, approaches to decomposition of heterogeneity using class encoding for class heterogeneity, and the Bidirectional Data Partitioning technique for feature space heterogeneity. Applicability of feature merit measures and data complexity measures for subproblem evaluation, and the related search strategies, is proven by the experimental results on the synthetic and real data.

This research is currently in the mainstream of major activities in data mining, machine learning, and knowledge discovery from large heterogeneous databases.

1.2 Thesis statement

Unstable feature relevance in classification tasks is the research problem being investigated. In this thesis, it is considered to be an expression of classification heterogeneity. Therefore, the problem is solved introducing the basic heterogeneity types and their variations. A classification problem is regarded to one of heterogeneity types based on prior domain knowledge, exploratory analysis, preliminary heterogeneity tests, or an assumption and its verification in case there is no other clue.

For *class heterogeneity*, it is assumed that a subset of relevant features differs in homogeneous regions that correspond to different classes, or subsets of classes. For *contextual heterogeneity*, it is assumed that there are contextual features that specify subproblems. If the contextual features are not available, and heterogeneity does not appear at the class level, which is *feature space heterogeneity*, it is assumed that heterogeneity presence can be identified exploring other data characteristics.

The main statement is that decomposition into subproblems representing homogeneous regions can effectively model heterogeneous classification problems. Decomposition is performed within an ensemble framework. It is expected that ensemble learning will help to improve predictive performance, while decomposition will help to reveal some structure or meaningful patterns in data.

The main goal of this research is to develop a general approach for all heterogeneity types, and suggest the solutions. This goal has subsidiary, more specific research goals. The first goal is to develop the theoretical background for the research problem. At this level, conceptual analytic research is applied to investigate types and variations of heterogeneity, data characteristics that can be associated with classification heterogeneity, and the benefits of combining ensemble generation methods. At this research stage, various theories, models, and frameworks applied in prior significant studies on the topic are considered,

and the problem is formulated using the basic concepts, definitions, terms, and notions.

The second goal is to propose practical solutions for the basic heterogeneity types. This includes investigation of known techniques and measures to be used as components of the suggested multi-model solutions, development of the technique called Bidirectional Data Partitioning (BDP) as a solution for feature space heterogeneity, and an empirical evaluation of applicability of the proposed techniques. At this level, constructive and experimental research approaches are applied.

The collection of data sets for the experimental study includes the benchmark data sets used by the data mining, pattern recognition and machine learning communities, the synthetic data sets representing different heterogeneity types and other data properties of interest, and the real medical and biomedical data sets in the field to cancer research.

The research results show that information-theoretic and geometrical data characteristics used with an appropriate search strategy are applicable to uncover data structure related to heterogeneity. The experimental results have demonstrated that the proposed decomposition approach and the derived BDP schemes perform better than a unimodal approach, and, by preliminary results, better than some state-of-the-art ensemble techniques in terms of classification accuracy. It was shown that in cancer survival analysis and in discovery of cancer subtypes, BDP has a potential to provide meaningful results.

1.3 Thesis overview

In this section, a brief thesis overview is provided. Chapter 2 is devoted to multi-model classification overlooking a traditional approach established in the data mining community. This chapter provides an overview of concepts and formalizes classification, clustering and feature selection tasks. The related basic algorithms are described in Appendices. Introductory material is provided on a relatively new perception of predictive problems accentuating data structure and intrinsic complexity of a classification problem given a data set as a marginal description of the observed phenomenon.

Ensemble learning is introduced as an established multi-model approach. The rationale for using an ensemble of predictive models to accomplish decomposition of classification problems is presented. Basic methods of ensemble generation and combination of learning models are discussed. Three major categories of ensemble methods are described in connection with the proposed bidirectional partitioning technique, which can be viewed as a combination of the three.

Chapter 3 introduces the classification heterogeneity problem that served as a motivation for developing the bidirectional partitioning technique. The problem is presented in theoretical generalized form. Variations of classification

heterogeneity are described. Data structure for different heterogeneity cases is given interpretation.

In particular, feature space heterogeneity is presented as unstable feature relevance. Related literature overview is provided. Chapter 3 also introduces three approaches to perform decomposition of heterogeneous classification problems into subproblems related to three different heterogeneity types. Search and evaluation, the two constituents of decomposition are discussed. Feature merit measures used to encompass the evaluation part are described. The ensemble technique combining local feature selection and class encoding is developed using a formalized decomposition scheme. Different integration strategies are outlined and a dynamic selection method of integration based on the probability estimates is detailed.

Chapter 4 details the decomposition approach of bidirectional data partitioning (BDP). This approach is introduced as optimization of class separability in local regions implemented by means of local feature weighting and clustering. Data partitioning via clustering at different levels of granularity with respect to class labels is described. Two component classifier integration schemes are presented. Practical implementation of BDP is detailed covering different BDP schemes.

Chapter 5 provides case studies for empirical evaluation of the proposed BDP technique and its multiple schemes. Implementation issues and related experimental settings are described. Experiments are carried out on synthetic and benchmark data sets. Class separability and complexity measures as possible candidates for BDP's evaluation function are studied. Evaluation of superclass/subclass structure with BDP is presented.

Chapter 6 describes real data sets and experiments with BDP in medical and biomedical domain. The related data pre-processing techniques are discussed. The results are given extensive interpretation.

Chapter 7 is the thesis summary, with conclusions, limitations, and prospective work. Background materials are included in the Appendices.

1.4 Contributions of the doctoral research

The main idea presented in this thesis is that heterogeneous classification problems, the origins of unstable feature relevance, can be decomposed into subproblems and approximated with a set of predictive models covering homogeneous regions. Decomposition of a heterogeneous classification problem to construct those models can be performed assuming presence of certain data characteristics based on which the classification problem can be related to a particular type of heterogeneity. In some cases, those characteristics are a part of domain knowledge, in other cases they can be uncovered using preliminary heterogeneity tests, or assumed and verified with respect to performance of the suggested techniques.

Author identifies three basic types of heterogeneous classification problems: class heterogeneity, contextual heterogeneity, and feature space heterogeneity. Variations of classification heterogeneity are defined based on these three basic types and their combinations.

Author suggests and discusses sources of classification heterogeneity in general and provides examples from medical and biomedical domains.

In the doctoral thesis, author has proposed a general solution for a case of feature space heterogeneity, which is the hardest due to absence of key information to perform decomposition. The technique implementing this solution is named Bidirectional Data Partitioning (BDP). Approach to a contextual heterogeneity decomposition suggested by the author is an evolved version of the solution described in early work of Apte *et al*. (1998), where heterogeneous classification problems were mentioned for the first time. An advanced solution for contextual heterogeneity is among author's topics for a future research. In Licentiate thesis that preceded doctoral research, author has suggested a solution for a simpler case of heterogeneity that appears at the class level, a class heterogeneity variation, which is refined in the doctoral thesis. In the doctoral thesis, author has shown that bidirectional partitioning approach is applicable for all types and variations of heterogeneity.

A solution developed in this thesis has been applied to cancer survival analysis and cancer genomics, and demonstrated encouraging results.

Out of this research, author has developed and implemented BDP as a meta-classifier in the open-source non-commercial data mining project, WEKA. Software implementing BDP will be improved and submitted for the next release of the WEKA system. Author has extended functionality of other modules in WEKA and added implementation of heterogeneity tests based on class separability and geometrical complexity measures creating a preliminary analysis component in WEKA.

Practical implementation of the proposed approach resulted in extended functionality of the BDP technique. Integration into WEKA system, among others, opened a possibility to perform integration of classifiers into ensemble in different ways and covering homogeneous regions in data with multiple models using WEKA's concept of meta-classifier, different clustering and feature selection techniques. With certain settings, BDP can also be reduced to functionality of COSA, a subspace clustering technique. However, a separate implementation of COSA in WEKA is planned. Many of the above possibilities are not explored in this thesis, but provide a solid background for further experiments.

The research work described in this thesis comes from the original author's research that has been performed without collaborators since publication of the Licentiate thesis in 2005. It has been presented in more than ten Int. scientific conferences and published in seven single-author papers referenced in this thesis. Some of those papers along with early papers co-authored with other researches are mentioned in the next section.

1.5 Summary of author's selected published works

During the years of research related to the topic of this thesis a number of research papers has been published. This summary highlights various aspects of the unstable feature relevance phenomenon described in these publications. The efforts toward developing a strategy for data decomposition and a multi-model solution based on the theory of ensembles led to exploration of alternative strategies that was not covered in the thesis. Some of them are mentioned below.

The most recent paper (Skrypnyk, 2011) is devoted to analysis of various class separability measures and their suitability as a criterion in bidirectional data partitioning as well as an independent characteristic of a problem domain with respect to heterogeneity presence. Easily separable classification problems or problems with globally relevant subset of features are not subject to feature space or class heterogeneity decomposition. Complexity measures, based on heuristics not directly related to Bayes minimum error rate and exploring geometrical properties of data are used to support the conclusions.

In Skrypnyk (2010) Bidirectional Data Partitioning (BDP) technique is explored with DBSCAN weighted distance-based clustering on entire data set with IPA-based merging procedure that joins subgroups in one go. The criterion used in that version of BDP is based on a difference in intra- and inter-class distances. Based on these results author have extended functionality of BDP adding clustering inside classes, agglomerative merging procedure, estimation of DBSCAN parameters inside classes and possibility to use weighted distance-based *k*-Means clustering. In this paper, local feature selection is performed using feature values overlap heuristics that has its limitations, and does not produce meaningful results in case of nominal values. Taking into account this fact, author has implemented a possibility to use an external feature selection technique in groups of instances, perform feature selection by means of feature weighting, or use a combination of both in current version of BDP. Experiments with several benchmark data sets from UCI repository were not particularly encouraging and it motivated additional evaluation of data characteristics in these data sets. It was established that these data sets are not suitable for heterogeneity decomposition. However, other data sets have shown accuracy improvement and were used in further analysis thereafter.

The paper by Skrypnyk (2008) mainly investigates feature weighting based on the feature values span in each dimension as a measure of dispersion for the ability to improve class discrimination in subspaces. It describes the details of entropy-based regularization and investigates weights adaptation in the local neighborhood using a case-based study. It also presents distance-based selection of a local model for new instances. Based on these results, current version of BDP has been supplied by a meta-classifier as an alternative method of local model selection. Neighborhood purification procedure is described in

order to cover for weight adaptation mistakes. Later, this led to experiments with the β parameter and introduction of clustering inside classes.

In Skrypnyk (2007), a BDP's prototype approach, Localized Selective Partitioning based on the intra-class and inter-class ratio criterion and weights based on feature values overlap is presented. An analytical solution cannot be directly obtained for this criterion and it provides a suboptimal solution.

Papers by Skrypnyk (2009) and Skrypnyk & Ho (2006) are devoted to the Stochastic Discrimination theory (Kleinberg, 1990) and the stochastic discrimination multi-model technique based on the coverage optimization paradigm. It has been concluded that stochastic component successfully used in many ensemble techniques is capable of boosting predictive accuracy, but has little potential in knowledge acquisition related to data structure required in such disciplines as cancer genomics. When feature relevance is not equally distributed among features, Random Subspace Method and stochastic discrimination are not competitive to other techniques (Skrypnyk & Ho, 2003). Research on stochastic discrimination is not included to this thesis.

Decomposition strategies related and not related to class labels are explored in (Skrypnyk, 2004; Skrypnyk, 2002a; Skrypnyk 2002b). Earlier works are motivated by exploration of ensemble techniques in combination with feature selection (Puuronen *et al*., 2001). Feature selection by means of ranked feature merits is used as a part of ensemble feature selection. In order to stabilize the results obtained by a cut-off threshold value, data driven adaptive generation of candidate features has been used to stabilize the results in a range of threshold values. Criterion for inclusion of additional feature candidates besides those already included from the top of rank is directly related to accuracy estimates. Favorable results on accuracy have been obtained. This and other research work of that period explored stability of feature relevance in subproblems. Different integration strategies, static and dynamic, selection and voting, have been preliminary tested with decomposition into ensemble based on locally relevant features in Tsymbal *et al.* (2001).

2 MULTI-MODEL APPROACH TO CLASSIFICATION

The tools provided by machine learning, such as generalization, induction, validation, bias considerations, are indispensable for knowledge discovery. Data mining methods are based on machine learning techniques along with statistical, pattern recognition, and other techniques.

Machine learning is a study of algorithms that automatically improve their performance with experience (Hall, 1999). Prediction is a central task of those algorithms. Building a model within the pattern representation is accomplished by learning. Learning is not related to the exact representation of the data, but to the process that generates the data. In other words, from the specific knowledge provided by domain examples, an inductive learning method is capable to obtain general domain knowledge. If the constructed model exhibits good generalization, it is likely to make good predictions for new data.

Usually data items called instances (objects, or examples) are represented as attribute-value pairs. In some tasks, a structured representation of the domain objects is more natural. The term feature is used for a formal view of the structured data representation. Structured representation means that an instance is represented by a set of features taking some values. Each feature is a particular dimension in which the instances viewed. Feature selection techniques derived from machine learning provide one of the best solutions for high-dimensional problems.

Research described in this thesis combines these important achievements of machine learning in developing the approach to decomposition of heterogeneous classification problems. This chapter introduces the basic concepts and definitions of supervised and unsupervised learning algorithms. Section 2.1 provides a formal description of the classification and clustering

tasks in a single framework along with the related task of feature selection and dimension reduction. The notion of feature relevance in classification is explained. Sources of classification problem complexity are given a special attention following by a closing subsection that considers performance evaluation of learning algorithms.

2.1 Classification and related tasks

Often in data analysis it is useful to consider dividing the set of instances into classes in a way that instances within a class are similar to one another. The *classification task* occurs in a wide range of human activities when some decision or forecast is made on the basis of currently available information, and a *classification procedure* (or classification rule) is then some formal method for repeatedly making such judgments in new situations. (Michie *et al*., 1994) Classification tasks can generally be handled relatively well with machine learning techniques (Halteren, 1999). In machine learning classification is performed as *inductive learning*, or *induction* that is generalization from the known to the unknown, so that appropriate responses to the unknown can be formulated when it appears. For example, to determine whether an animal is a giraffe people know to look for dark patches and horns rather than estimate its tail or ears. Thus, patches and horns form the concept (generalization) of a giraffe. Now the unknown new animal, which is a leopard having dark patches also, cannot be assigned to the class "giraffe", because it doesn't have horns.

Statistical approach is generally characterized by having an explicit underlying probability model, which provides a probability of being in each class. Commonly, statistical classification models provide an estimate of the joint distribution of the features within each class, which in turn provides a classification rule. (Michie *et al*., 1994)

Classification methodology has been applied in many diverse disciplines. In statistics, as well as in the applied fields, such as pattern recognition, it is referred to as classification. In machine learning the corresponding term is supervised learning. Data mining encompassing both, statistical and machine learning techniques, relate classification to the prediction tasks along with regression (Weiss *et al*., 1998).

In this thesis classification is considered as a task of predictive data mining. The respective terms will be used throughout the text, borrowing when appropriate the terms from machine learning, statistical decision theory and information theory.

2.1.1 The classification task

Classification tasks in data mining are presented as specific goals, which are related to the instances with known class labels to be used in construction of a predictive model in order to assign class labels to the new instances. Thus,

instances with known class labels should be available. Each new instance must be assigned to one of a set of pre-defined classes based on observed instance descriptors, *features* (or attributes). For the above example of animal classification horns, patches, legs length and neck length are some of the descriptive features.

The *classification task* is to construct a procedure that will be applied to a sequence of instances in which each new instance must be assigned to one class of a set of pre-defined classes based on observed features. Classification produces categorical class labels unlike regression that models continuous-valued function.

A learning algorithm (or induction algorithm) forms the concept description inducing some general function from the specific example data called the training set, or a set of training instances. In concept learning, the most studied machine learning approach, a target concept is a function over instances according to which class labels are assigned according to the underlying distribution. Concept description is a model (hypothesis, or knowledge) that the learning algorithm has induced from the data. This model for classification task takes a form of some discrete function, which hypothesizes, or estimate, the true value of a class variable.

Suppose some functional dependence $y = g(\mathrm{x})$ exists between features x and a class variable y that is exemplified by the training instances. Approximation of this functional dependence $\hat{y} = h(\mathrm{x})$ is built by a learning algorithm L using the training set TR. As a result, a certain release of this model called a classifier C is produced. The training set TR is a set of training instances $\{(\mathrm{x}_1, y_1), \dots, (\mathrm{x}_M, y_M)\}$, where x_i are vectors of the form $\langle x_{i,1}, \dots, x_{i,j}, \dots, x_{i,N} \rangle$, where $x_{i,j}$ are feature values of x_i, and M is the size of the training set TR. Features will be further referred to as variables $f_j, j = 1 \dots N$, where N is the number of features. Class label y_i is a special categorical feature taking its values within the range delimited by the number of classes, $y_i \in (c_1, \dots, c_d, \dots, c_D)$, where c_d are the class values, $d = 1 \dots D$. Other features may take either categorical or numerical values which might be discrete or continuous.

Classification is a two-step process. The first step is model construction, or training. At this step, the training set TR containing instances (x_i, y_i) is used by learning algorithm L to produce a model $\hat{y} = h(\mathrm{x})$ presented by classification rule, decision tree, or analytical expression.

The second step is model usage, or application. During this step model accuracy is estimated on the test set TE having the same representation as the training set. Test set should be independent of the training set in order to have more reliable estimates. The data can be partitioned onto the training and test sets before model construction. The details are considered in Section 2.3.

For each instance from the test set (x_j, y_j) the estimate $\hat{y}_j$ from the model is obtained and compared with the known class value y_j. Then the accuracy rate or the error rate is calculated. The accuracy rate is the percentage of the instances from the test set that are correctly classified by the model.

Correspondingly, the error rate is the percentage of the instances from the test set that are incorrectly classified by the model. The other measures for evaluation of the constructed model, that is performance of the learning algorithm, which is used to construct the model, are considered in Section 2.3.

A classification problem and its solution can be described in the terms borrowed from decision theory, such as logical decision rule, and from statistical decision theory, such as class boundary, decision boundary and decision surface (or decision regions). Training instances can be schematically represented as points in some space or probability distribution, for example, in the feature space or its projection. Then class boundaries describe location of the training instances of different classes in this space, the structure of data. Decision surface demonstrates coverage of the data structure by a predictive model constructed, i.e. by a classifier.

In order to view and comprehend a phenomenon a multidimensional data representation comes handy in data mining. In classification and clustering tasks data instances are considered as points in multidimensional space, where axes are features or attributes. Multidimensional presentation of data from practical classification tasks shows some structure that typically differs from data created by random processes. For example, an image data represent an important category of structured data. For image data processing typical predictive methods involve split-and-merge approaches (Starck *et al.*, 1998). Instances in image data are pixels or regions of the image. Decomposition of an image is a part of split-and-merge – an image is successively divided into smaller regions until a homogeneity criterion is met. A homogeneity criterion can be based on the pixel values or grey-levels within the corresponding image region.

2.1.2 The clustering task

Clustering, or unsupervised learning, is applied in data mining in order to discover unknown categories or groups in data based on similarity or dissimilarity concepts. Clustering can be useful in finding structure in data, for example, hierarchical relations between categories. Clustering results are often evaluated by known class labels, but this approach is not always applicable.

The idea of using clustering as a supplement to classification has been marginally used in pattern recognition community since early 1960's. However, systematic approaches to combination of clustering and classification appeared only in early 90's along with research on local learning and ensemble classification (Fradkin, 2006).

Clustering methods differ in the assumptions about the nature of data, constrains applied to data partitioning, and similarity measures. Therefore, different methods may lead to different results, which cannot be directly compared, because the choice of criterion for comparison is not straightforward as in the classification task. Thus, the choice of clustering method depends on the nature of data and the researcher's goal.

Clustering task is often viewed as an optimization task. Therefore, clustering algorithms are often computationally demanding. Finding clusters in high-dimensional data is particularly challenging. Another challenge faced by clustering is finding arbitrary regions in data of uneven density and shapes with possible noise and outliers.

Various data analysis techniques have in common the intent to analyze, summarize and extract useful information from data. Many of them consider data in a form of a rectangular table and operate simultaneously by two sets, a set of instances (examples, vectors) and a set of features (attributes, variables).

A wide variety of techniques are based on simplifying data matrices performing a bidirectional search. In data mining, these techniques originate from factor analysis and clustering methods. Their procedures of bidirectional partitioning differ in the assumptions they rely on, data types to which they apply, and the criteria reflecting a data analysis task.

In clustering, a bidirectional data partitioning approach, also known as subspace clustering, two-mode partitioning, and block clustering, has been actively researched over a few decades (Rosmalen *et al.*, 2009; Madeira & Oliveira, 2004; Kriegel *et al.*, 2009; Domencioni *et al.*, 2004; Parsons *et al.*, 2004; Moise & Sander, 2008; Mechelen *et al.*, 2004). Clustering techniques of this type seek a set of instances assigned to a cluster along with a subset of features as the related dimensions. A search is performed simultaneously in the subspace of instances and in the subspace of features in order to partition data. In other words, both rows and columns of a data table are assigned to one or more clusters. Elements in the same cluster are close to each other in terms of a pre-defined distance function or a similarity measure. However, an evaluation function does not rely on class labels. This thesis discusses a technique based on a bidirectional data partitioning methodology transferred to classification tasks, which follows an ever-growing demand in effective ways to process contemporary data.

A problem of unstable feature relevance in classification (Lazarevič *et al.*, 2000) has become a main motivation for the proposed technique. Unstable feature relevance is a distinguishing characteristic of heterogeneous classification problems (Apte *et al.*, 1998). From a classification task perspective, one may assume existence of "regions" in data where a true data structure is presented by a smaller subset of locally relevant features. Disregarding irrelevant features provides a clarified data structure. This description implies that normality of class distribution used as an assumption in many statistical measures is violated on the entire data set. But in those "local regions" it may hold true. The goal is to identify those regions in case they exist.

A solution to this problem can be obtained by solving an optimization task. This involves a simultaneous search for subsets of features and subsets of instances with an optimum in a criterion function that in a way reflects "stabilized feature relevance". The next section briefly outlines an optimization task.

Clustering finds groups of instances in data identical in some sense. It resembles discovering classes, which are unknown. Thus, often the existing class labels are used to evaluate clustering results. In real world, division by classes sometimes can be uncertain. The more relevant information is encoded in data, the better is classification. The lack of relevant information presented by features, along with limited domain knowledge motivates, seeking data structure beyond class labels, for example, superclasses and subclasses. On the other side, classes could be well known, but information encoded in the descriptive features may not be enough to describe them without ambiguity. Supervised, semi-supervised, and unsupervised learning all have related goals and related tools. Therefore, class separability measures can be adopted from measures used in clustering.

2.1.3 Relevance of features and feature selection

For classification problems importance and contribution of a feature to classify instances, i.e. its predictive ability, is usually expressed by a degree of feature *relevance*. In some cases, it is convenient to project feature relevance on a gradual scale and consider a degree of feature relevance. In other cases, it is sufficient to consider two extreme meanings, *relevant* or *irrelevant*.

For prediction tasks, the notion of relevance is related to both features and instances (Blum & Langley, 1997; John *et al*. 1994). Different definitions of relevance depend on the particular goals and related to the question "relevant to what?" Probabilistic definition of feature relevance is proposed in Kohavi (1994). Relevance to the target concept (according to machine learning definition), relevance to the sample/distribution (according to statistical definition), relevance as a complexity measure and relevance as incremental usefulness are considered in Blum and Langley (1997).

Besides being *relevant* or *irrelevant*, features can be *redundant* and *interacting*. In this thesis, the following definitions are used. Features that provide information about the class for a given set/subset of instances are called *relevant* features. Features that do not provide information about the class for a given set/subset of instances are *irrelevant*. *Interacting* features are those whose values are dependent both on the values of other features and on the class variable for a given set/subset of instances. One of two interacting features may be discarded if it does not imply any loss of information about the class. *Redundant* features are those whose values are dependent on the values of other features regardless the class for a given set/subset of instances, that is they may be created as a transformation of a relevant feature, and as such, provide no further information about the class.

In spite of the seeming conclusion that redundant features in a high-dimensional data could contribute to learning improvement by additional information, machine learning generally points to the contrary. Increasing the number of dimensions leads to exponential growth of the data quantity needed for reliable learning.

The usefulness of redundant features depends on the complexity of a classifier and on the proportion of instances in the training set to the number of features so that more simple classifiers perform better in high-dimensional classification problems than more complex ones, which require more parameters to estimate from the training set. Construction of a learning model considering redundant features might be useful in very specific cases (Skurichina, 2001).

Two features are called interacting if dependence between the class variable and a feature is conditioned by the values of another feature. The level of interaction may vary. The following example of conditional dependency is taken from the problem of prediction of automobile accidence risk. A feature "driver's age" taking value "17-23 years" acquires great significance if, and only if, a feature "sex" takes value "male" at the same time. In Hall (1999) features are considered under moderate level of interaction if they are individually predictive of class at least some of the time. Features whose ability to predict the class is always dependent on the others exhibit higher order dependencies.

In the vast majority of classification problems contribution of different features for predictability of classes (class discrimination) is not equal. Usually relevant features are unknown prior to learning. Also, when a data set contains too many features, a practical need arises to select a relevant subset of features for generating a model for classification.

Feature relevance is estimated using feature merit measures. There are measures designed to evaluate an individual contribution of a feature to discriminate between classes / predict classes, contribution of a feature considering interaction with other features, or contribution of a feature subset. *Feature selection* techniques mostly based on individual feature / feature subset measures. The aim of feature selection is to choose a subset of features in order to improve prediction accuracy and/or simplify a classifier without significantly decreasing prediction accuracy by means of building that classifier using the selected features only. There are also several other definitions considered in Dash and Liu (1997) looking at the feature selection task from various points of view.

Majority of classification methods are based on evaluation of features contribution to discriminate between classes rather than on the intrinsic data characteristics. The examples are decision tree, Naïve Bayes and *k*-Nearest Neighbor learning algorithms. However, the embedded feature selection is not always effective. The curse of dimensionality problem may arise. In particular, when the model is built over many features it becomes large and hard to interpret. The basic learning algorithms with embedded feature selection usually evaluate features individually, thus they are not capable to identify feature interactions and redundant features. A partial solution for this problem is provided, for example, by the random subspace methods (Ho, 1998; Skurichina & Duin, 2001).

Contribution of a feature to discriminate between classes can be measured as (1) the difference between the prior uncertainty and expected posterior

uncertainty using this feature (the Information gain from a feature) (Soofi, 2000; Quinlan, 1986), (2) the ability to predict a class variable from the considered feature (correlation between a feature and a class that may be based on distance or Information gain measures) (Hall, 1999), (3) divergence or distance-based separability, when one feature is preferred to another if this feature induces a greater difference between two-class-conditional probabilities than another feature (Ho, 2002).

Some measures that evaluate the worthiness of feature subsets take into account feature dependencies without finding the explicit form of the dependency. The simplest method of feature subset evaluation performs evaluation of randomly selected subsets iteratively. Usually such feature subset selection methods use performance of the particular classifier as an evaluation function, which produces biased estimates. The example is a classifier error rate measure used in wrapper feature selection (Kohavi & John, 1998).

Feature selection and dimension reduction is also performed in unsupervised learning. Unsupervised feature selection methods analyze only intrinsic characteristics of data such as variance of feature values. Contrary, supervised filter techniques assess relevance of features evaluating data characteristics related to distributions of classes.

2.1.4 Sources of classification complexity

Classification problem complexity is often associated with geometrical complexity of decision boundary or class boundaries. Recent studies on complexity measures (Ho *et al.*, 2006) have shown that boundaries between classes, not class shapes, contribute to problem complexity and, hence, classification performance of different classifiers. Wider margins between classes reduce demand on the precision of decision boundary.

A widely used "divide and conquer" approach can be applied to reduce complexity of a classification problem, while class separability measures can serve as a criterion for decomposition into subproblems. Presence of irrelevant features is an important constituent of complexity. Although the idea of reducing complexity by means of improved class discrimination is central to many data mining techniques, including clustering, feature selection, extraction and discretization, in this study it is explored for potential application in two-way decomposition schemes that combine clustering and feature selection.

Practical classification problems are created by non-chaotic processes with some underlying physical or behavioral models (Ho & Basu, 2002). These processes create data with some distinctive structure even in presence of some stochastic components (Ho & Basu, 2000). However, class labels incorporate information strongly biased by a human perception of the phenomenon and cannot be easily obtained in some situations (Xiang *et al.*, 2008).

Classification problem can be difficult for different reasons. There could be an intrinsic class ambiguity due to specifics of a problem and features chosen to represent it. In this case there is no possible improvement beyond a certain point. Some problems known to have nonzero Bayes error, hence the classes are

ambiguous regardless of sample size or feature dimensionality. On the other hand, the class discrimination problem can be difficult due to decision boundary complexity that can be reduced. For example, structure related to subclasses may impose a complex decision boundary. Problem presented by sparse data may appear deceptively simple (Ho & Basu, 2002)

Analysis of data structure can be help improving classification accuracy.

Geometrical properties of high dimensional data in classification and clustering are studied by Jimenz and Landgrebe (1998). Their research shows that human perception of three-dimensional space is not applicable to understanding geometrical and structure statistical properties of data in higher dimensions. It tends to mislead one's intuition when it comes to the choice of data analysis methods. Using Euclidean and Cartesian geometry, they provide a mathematical proof that leads to the following conclusions.

High dimensional space is mostly empty, which implies that multivariate data in R^N is usually in a lower dimensional structure. As a consequence, high dimensional data can be projected to a lower dimensional subspace without losing significant information on class separability and data structure.

Data instances in Gaussian distributions will have a tendency to concentrate in the tails. In uniform distributions data instances will more likely reside in the corners, making density estimation more difficult. Local neighborhoods of a fixed radius are mostly empty, requiring the radius to be large in order to capture instance. It produces the effect of losing detailed density estimation and leads to data sparsity.

These findings provide an explanation why the stochastic discrimination method fails in high dimensions without random subspace projections involved (Skrypnyk & Ho, 2006; Skrypnyk, 2009). Interpreting unstable feature relevance in machine learning as a mixture of distributions in high dimensions helps to understand functionality of distance-based subspace clustering and bidirectional partitioning studied in this thesis.

Authors in Jimenz and Landgrebe (1998) find support for the aforementioned tendency in the statistical behavior of normally and uniformly distributed multivariate data at high dimensionality. It is expected that as the dimensionality increases the data will concentrate in "an outside shell". As the number of dimensions increases that shell will increase its distance from the origin as well.

Performance of classification and clustering techniques has been linked to geometrical properties and complexity of data only recently (Singh *et al*., 2002; Pranckeviciene *et al*., 2006; Bernadó-Mansilla & Ho, 2005). Acknowledgement of this fact is important in understanding the sources of classification problems complexity and the phenomenon of feature space heterogeneity.

2.2 Learning and prediction using multiple models

Decomposition of the classification problem stated on heterogeneous data

represents each homogeneous region as modeled separately in a particular subspace of relevant features. It can be presented using an ensemble framework based on multiple learning models. The component classifiers of an ensemble can be built using methods that combine local feature selection, class encoding and sampling.

Combining multiple learning models into ensemble is based on many theoretical reasons and an empirical evidence of the effectiveness. This section introduces ensemble learning as to a separate machine learning direction and reviews these theoretical reasons and the obtained empirical results.

An ensemble learning framework and a rationale of using multiple models are addressed in Subsection 2.3.1. Ensemble methods based on local feature selection / feature set manipulation, class encoding, sampling and their combination are presented in Subsection 2.3.2. The results obtained with these methods are briefly reviewed. The combined methods are accented in connection with decomposition of heterogeneous data.

2.2.1 Ensemble learning: a general framework

Multi-model solutions are widely used in machine learning, data mining and pattern recognition. Usually, the final solution is derived out of consensus between the component models. A plenty of terms are used in the literature: ensemble learning, decision committee, ensembles of learning machines, ensemble of learning models, classifier fusion/combination/aggregation, or multiple classifier systems. The terminology reflects a particular way to integrate multiple models or specific classifiers used to build the component models. In this subsection, the basic notions with respect to multi-model approach are introduced.

In this study, a multi-model solution will be considered as an *ensemble of classifiers* as a set of models. In Subsection 2.1.1, a notion of a classifier has been introduced as a particular realization of a learning model. Given the training set TR and a single learning algorithm L the model $\hat{y} = h(\mathrm{x})$ can be constructed in different ways. Multiple models $\hat{y}_i = h_i(\mathrm{x})$ are collected and integrated into an *ensemble*. The way to produce those multiple models, i.e. multiple classifiers, is a subject of the *ensemble generation* techniques. Multiple classifiers that compose an ensemble are called *component* classifiers. Sometimes in the literature, component classifiers are referred to as base, constituent or individual classifiers. In this thesis, the *base* classifier refers to a particular learning algorithm, such as J48 decision tree or 1-Nearest Neighbor. The *component* classifier refers to a model built within an ensemble, a member of ensemble created using a base classifier. The way to integrate (combine or select) multiple predictions obtained from the component classifiers is a subject of the *ensemble integration* techniques.

Let us denote the component classifiers as $h_1, \dots, h_s, \dots, h_S$, where S is the size of an ensemble. An *ensemble of classifiers* is a set of learning models (component classifiers) whose individual decisions are combined in some way to classify new instances. Commonly, prediction of class membership for a new

instance obtained from ensemble of classifiers is viewed as a two-stage process that includes *learning* and *prediction* (Figure 1).

During the learning stage, learning model construction over the initial training set TR is performed multiple times in different ways to obtain multiple models. It is usually done by modifying TR (for example, sampling multiple times, encoding multiple times, or partitioning at once) and altering the model generation process. Predictions of the component classifiers are integrated in a certain way F to derive $h^* = F(h_1, \ldots, h_S, \ldots, h_S)$ that will be used to obtain a final prediction. At the prediction stage, a new instance $(\mathrm{x}, ?)$ with the unknown value y is given as an input an ensemble and class value y^* is predicted as $y^* = h^*(\mathrm{x})$. *Generation* and *integration* of the component classifiers are two key steps of ensemble construction. Many existing categorizations of ensemble techniques are mostly based on the differences in generation and integration.

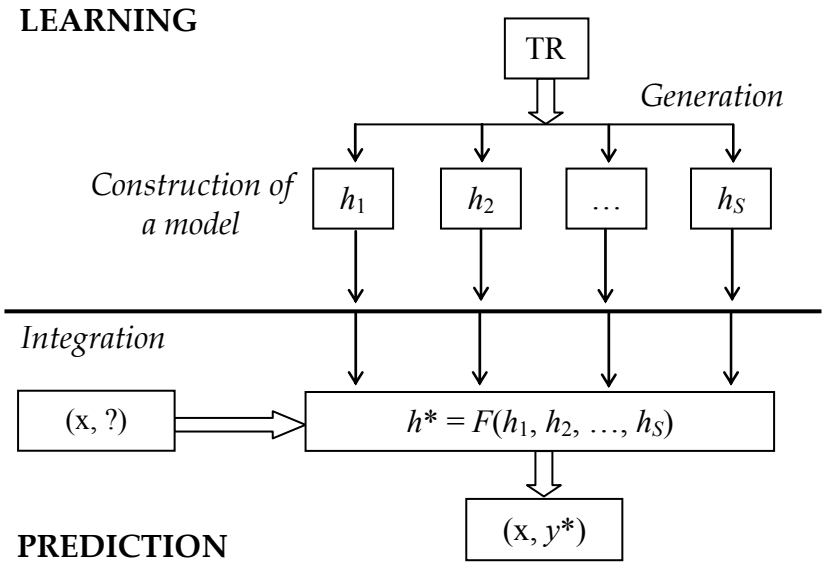

FIGURE 1 Ensemble of classifiers: learning and prediction.

Generation of the component classifiers is a subject of altering the training process, which is mainly derived from modification of TR. Modification is performed manipulating the set of training instances, the set features, class labels, or combinations of the above according to a certain sophisticated strategy. The strategy is based on an underlying theory, prior assumption, may involve search heuristics with evaluation function or a stochastic process. Further in the text, it will be referred to as an *ensemble generation rule*. The simplest approach to generate component classifiers is to produce alternative perturbations of the training process created by random sampling or random feature subspace projections. This approach can be used to optimize coverage of the data with multiple models picking models from the stream of randomly generated models (*coverage optimization*). Another approach is to partition data

and model parts separately. In this case the number of models is significantly smaller and their integration can optimize their predictions (*decision optimization*).

A new model in ensemble can be constructed independently of the other models or taking into account what models have been generated so far. For example, additive models like boosting, seek to minimize some criterion function that is based on training error. Figure 2 represents the process of component classifiers generation. In general, the ensemble construction process is iterative, that is generation of a new classifier depends on the current ensemble consistency. In some methods the inputs for all component classifiers are calculated at once, skipping the blocks *5*, 7, 9, and *10* in Figure 2. The process starts from initialization of the learning parameters (block *1*), then proceeds with setting a heuristic rule (block *2*) for a subsequent modification of the training set (block *3*). The block *3* outputs a training set TR modified by multiple feature subspace projection, class encoding, sampling, and so on. With this input a new classifier is generated (block *4*) and evaluated (block *5*). Subsequent modification of the ensemble (block *6*) includes addition and/or deletion of some classifier from/to the ensemble. Having the current set of the component classifiers, ensemble characteristics (accuracy, diversity, and so on) are estimated (blocks *7*) and then used for the stopping criterion evaluation (block *8*). Block *9* is a decision block altering the process: construction of the next component classifier (dashed arrow to the block *3*), modification of the ensemble generation rule (block *10*), or stopping the process with the current ensemble (block *11*).

Existing ensemble generation methods do not necessarily include all steps shown in Figure 2. For example, bagging (Breiman, 1996), which is based on sampling, generates a set of the component classifiers at once without their evaluation and modification of an ensemble generation rule. Thus, blocks *5*, *7*, and *10* with the corresponding outputs are not in use for bagging. Boosting (Freud & Schapire, 1996) evaluates each new generated classifier and changes the ensemble generation rule each time after a new classifier is added to an ensemble, skipping block *7*.

The rationale of using multiple classifiers in an ensemble is the following. Different component classifiers make errors for different instances due to their different design. When they are combined, they produce more accurate prediction comparatively to a single classifier trained to reach the highest accuracy for all instances in a data set. There are different theories explaining ensemble efficiency (Dietterich, 1997; Kleinberg, 1990; Schapire *et al.*, 1998). Usually ensembles are used to improve the prediction accuracy by mechanisms motivated by the learning theory rather than by particular data characteristics.

Many researchers (Ali & Pazzani, 1996; Dietterich, 1997; Maclin & Opitz, 1997; Opitz & Maclin, 1999; Tumer & Ghosh, 1996) has concluded that combining outputs of several classifiers can be useful only if they produce uncorrelated errors, that is if they are independent in production of their errors. When all component classifiers are identical an ensemble has no gain.

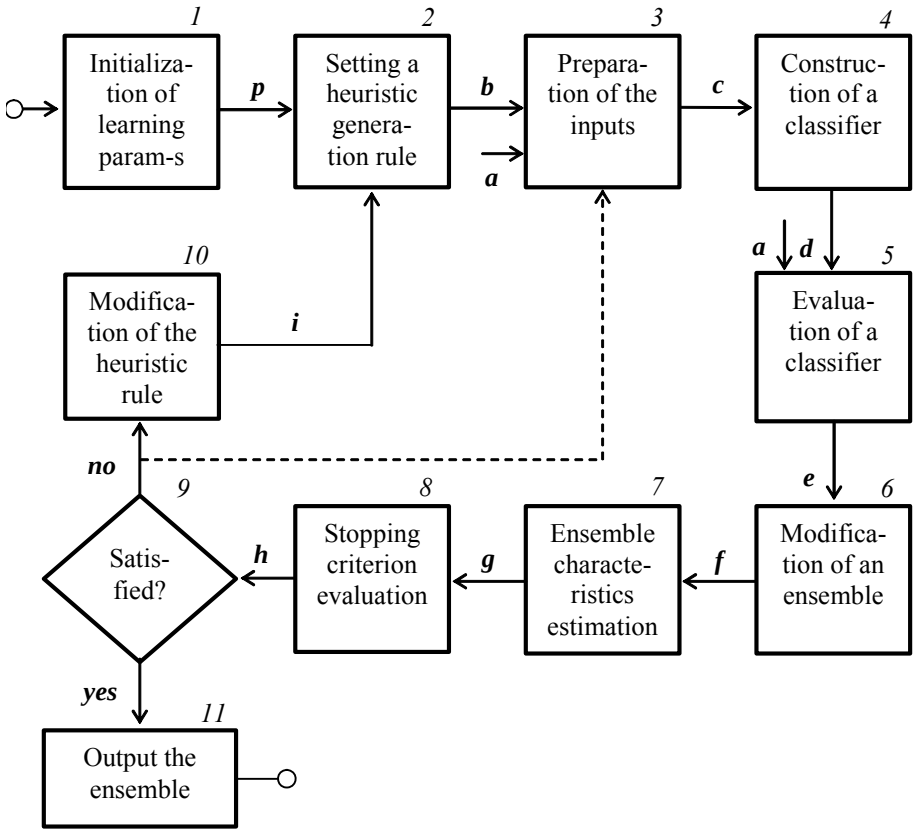

FIGURE 2 Generation of component classifiers. Block symbols represent the basic process stages, and small letters associated with arrows denote the input/output data for each stage. Real techniques does not necessary include all those components and arrows. Small letters near the arrows denote the following: ***p*** - learning parameters, ***a*** – a training set TR, ***b*** – a heuristic ensemble generation rule, ***c*** – inputs for the classifier generation (a modified training set), ***d*** – classifier, ***e*** – fitness value for a classifier, ***f*** – current set of classifiers in an ensemble, ***g*** – control ensemble characteristics, ***h*** – control value for the stopping criterion, ***i*** – parameters for a heuristic ensemble generation rule.

The Stochastic Discrimination theory (Kleinberg, 1990) considers supervised learning from the perspective of enforcing the uniform coverage of data by the set of models and suggests that a set of classifiers in ensemble should cover all instances uniformly and with equal chance of success. Then discrimination between classes is achieved when the number of the component classifiers is large and all classifiers satisfy the particular requirement of minimal difference in inclusion of instances from different classes. This theory, in particular, was successfully applied to explain performance of ensemble learning techniques such as boosting and the random subspace method (Kleinberg, 2000). The considered approaches emphasize properties of the component classifiers that will be integrated into ensemble. By the result of numerous ensemble studies

summarized in Dietterich (1997), error rate of each component classifier should not exceed 0.5. Otherwise, ensemble error rate will usually increase as a result of combination of their predictions. In particular, it has been shown that ensembles consisting of *weak* classifiers that make independent errors provide dramatic improvements in the predictive performance. Hence, the key to successful ensemble creation is to construct component classifiers that will be more than 50% accurate and will produce their errors independently.

However, the above argumentations about the properties that component classifiers should possess are strongly related to the integration strategy. For example, error correlation of the component classifiers and error reduction in ensemble is proven mathematically in Tumer and Ghosh (1996a) and Tumer and Ghosh (1996b) for integration by averaging. Independence of the component classifiers can be estimated using various ensemble *diversity* measures as considered, for example, in Cunningham and Carney (2000), Kuncheva and Whitaker (2002), and Prodromidis and Stolfo (1998). Stochastic discrimination theory is also applicable to *coverage optimization* techniques.

With respect to ensemble generation, there are three major categories of generation methods (Dietterich 1997): (1) *multiple feature subspace projection*, (2) *class encoding*, and (3) *sampling* the training set (changing the class distribution). Coverage optimization techniques are based on random perturbation or manipulation in (1), (2), (3), or in combination of those. In coverage optimization, the goal is to create a diverse set of classifiers with a given combination function. There are also decision optimization techniques that use probabilistic and Bayesian methods as well as majority vote, sum or product as a function of the component predictions, or assign scores/ weights to the component predictions to produce h^*. In decision optimization, the goal is to optimize some combination function for a particular set of classifiers. Bidirectional data partitioning technique and class encoding with feature selection schemes developed in this thesis, are examples of a decision optimization ensemble.

Good performance of an ensemble technique is achieved upon successful choice of generation and integration methods along with a learning model. There are stable and unstable classifiers with respect to their response to a change in the input data induced by the generation method.

Ensemble performance depends on the amount of correlation between predictions of the component classifiers and on integration of those predictions, which is directly related to the amount of classification error reduction in ensemble (Tumer & Ghosh, 1996b). Diversity and uniform coverage provided by ensemble generation method are not themselves a sufficient condition for good ensemble performance. The integration method chosen must take advantage of both diversity and coverage of the component classifiers.

Integration of the component classifiers is based on decision which of their predictions to accept as a final prediction or how to combine individual predictions to develop a final prediction. The corresponding strategies, *selection* and *combination*, are the basic strategies of the component classifiers integration.

An example of integration by selection is the cross-validation majority technique (CVM) (Schaffer, 1993; Kohavi, 1995). The commonly used combining method is voting, which is based on the majority voting principle (Bauer & Kohavi, 1999). Another basic combination method is *stacked generalization* (or *stacking*) (Wolpert, 1992). The integration methods used for in experimental sections of this thesis cover both categories: selection (based on distance and based on a meta-classifier in bidirectional partitioning) and combination (a variation of the stacking technique in class encoding).

There are two approaches to select or combine classifiers, *static* and *dynamic*. Static techniques analyze the outputs of all component classifiers and then develop a final prediction. They attempt to find a single "best" classifier for the entire data set. This approach, however, is not suitable for heterogeneous data. According to a dynamic approach the integration rule is revised for every unclassified instance: the most suitable of the component classifiers is assigned, or a combined classifier is built. The most suitable classifier is usually determined using some distance metric. An extensive review of different integration methods can be found, for example, in Tsymbal (2002). Different integration strategies, static and dynamic, selection and voting, have been evaluated with combined ensemble technique in author's joint work in Tsymbal *et al.* (2001).

Classification by ensemble provides accuracy increase due to disagreements of the component classifiers in different situations. Each component classifier is competent in certain situations, but domains of competence of different classifiers may overlap, making majority voting, and in particular, dynamic voting applicable. An attempt to relate domains of competence of the component classifiers and geometrical complexity of a classification problem has been made in Ho *et al*. (2006).

The theoretical backgrounds of ensemble learning are important to understand in order support argumentations for using ensemble framework while developing a solution for heterogeneous classification problems.

2.2.2 Feature set manipulation, sampling and class encoding

Methods belonging to three main ensemble generation categories can be used for heterogeneity decomposition only partially. Manipulation by the set of features in order to make multiple feature subspace projections can be performed as local feature selection. Class encoding provides a partition of the set of instances in accordance to class labeling that may be used for decomposition of class heterogeneity. Sampling provides a partition of the set of instances according to some heuristic rule in order to find instances representing homogeneous regions. The basic methods are based on the simple heuristic rules of ensemble generation, such as manipulating features, class labels, or training instances.

Constructing component classifiers in different feature space projections, i.e. on different feature subsets, has been proved to increase the diversity of an ensemble providing uncorrelated errors in predictions of the component

classifiers in general. However, success of this technique mainly depends on the classification problem specifics. Ensembles built manipulating features, contrary to sampling and class encoding techniques, keep the distribution of the training set unchanged. Pseudo-random multiple feature subspace projection methods are based on heuristic search and evaluation of different feature subsets for construction of the component classifiers towards effective ensemble prediction. The examples are the Random Subspace Method for decision tree ensembles (Ho, 1998), the Input Decimation method (Tumer & Ghosh, 1996b), and the Stochastic Attribute Selection Committees method (Zheng & Webb, 1998).

In the Random Subspace Method (RSM) for decision tree ensembles (Ho, 1998) the component classifiers are constructed systematically selecting feature subsets in a pseudo-random fashion. This way a random feature subspace in the original feature space is obtained, and a component classifier is constructed in this subspace. It was indicated that RSM is more successful when the information is uniformly spread over all features rather than it is condensed in few features, especially for small random subsets. When all features are informative, redundancy does not affect RSM performance.

Skurichina and Duin (2001) have shown that bagging performs better than RSM for the highly redundant feature spaces when the discrimination power is condensed in few features and the training set is small. However, RSM outperforms bagging when the discriminating power is distributed over all features.

Kuncheva and Whitaker (2001) investigated the potential improvement from ensembles constructed on feature subsets comparatively to a single classifier. They analyzed distribution and the extremes of improvement (or failure), the chances that ensemble outperforms a single best classifier when the feature space is partitioned at random, relationship between the spread of accuracies of the component classifier and ensemble, and the performance of integration schemes.

A genetic search for different feature subsets on which an ensemble of neural networks is constructed is considered in Opitz (1999). In order to optimize the ensemble characteristics further search for feature subsets is conducted. For a stopping criterion, a trade-off between ensemble diversity and accuracy is used.

The Input Decimation method (Tumer & Ghosh, 1996b) reduces correlation among errors produced by the component classifiers. In this method, features to construct the component classifiers are selected according to their correlation with a class variable. Oza and Tumer (1999) have adapted a correlation based approach to construct an ensemble of classifiers. To estimate the goodness of feature subsets correlation between features and a particular class was calculated, and features with highest correlation were selected to construct a classifier for that class separately producing different models for different classes.

Oza and Tumer (2001) have found that deleting even a few of the input features hurt the performance of the component classifiers so that accuracy of the voted ensemble degrades. The input decimation method performs well when the input features are highly redundant.

The Stochastic Attribute Selection Committees (SASC) method (Zheng & Webb, 1998) generates ensembles of classifiers stochastically modifying the set of features. It was concluded that on average SASC is more accurate than bagging and less accurate than boosting, but like bagging, SASC is more stable than boosting. Their work demonstrates a competitive strength of ensembles constructed by feature set manipulation compared to the most popular sampling methods.

Methods encoding a class variable were originally developed to solve multi-class problems. They are based on splitting a multi-class problem into a series of independent two-class problems according to the rule of class re-labeling called *dichotomizer* and recomposing them using dichotomizer's outputs. This approach is implemented, for example, in support vector machines, multi-layer perceptrons, and most rule-based learning algorithms (Masulli & Valentini, 2000).

Class encoding ensemble techniques are well-represented by three techniques described and compared in Masulli and Valentini, (2000): one-per-class, pairwise, and error correcting output codes (ECOC) ensemble techniques.

The one-per-class ensemble technique (Anand *et al*. 1995), also called in the literature one-against-all binarization, is designed to separate a single class from all the others. As a consequence, the number of the component classifiers in ensembles S equals to the number of classes D. Integration of the component classifiers is usually performed using some similarity measure. Binary outputs of all component classifiers are collected as a vector $(y_1, \dots, y_S)$, where $S = D$, and compared with the binary D-bit codeword corresponding to the class re-labeling scheme.

The pairwise ensemble technique (Hastie & Tibshirani, 1998; Moreira & Mayoraz, 1998, Fürnkranz, 2002), also called pairwise coupling or round robin, converts a D-class problem into a series of two-class problems by learning one classifier for each pair of classes using only training instances for these two classes and ignoring all the others. Ensemble consists of $S = D(D-1)/2$ component classifiers in this case. A typical integration scheme for the pairwise ensemble technique that works in many cases is simple voting (Fürnkranz, 2002). However, in this case for each new instance to be classified the outputs of majority classifiers are not significant and introduce noise for the voting scheme by the inappropriate information.

Error correcting codes are applied in the ECOC technique introduced in Dietterich and Bakiri (1991 and 1995), which is established as one of the most successful ensemble techniques in general. Correcting schemes taken from coding theory induce some redundancy in representation of subproblems that increases classification accuracy.

Ensemble techniques based on training set sampling have been recently under intensive research. In this category of ensembles, bagging and boosting are the best-known and most widely applicable techniques (Opitz & Maclin, 1999). The main principle of ensemble generation using sampling is the following. A number of folders /samples is obtained from the training set, and then the learning algorithm runs several times, each time on a different sample from the training set. Then, multiple classifiers are integrated to produce a final prediction.

Ensembles generated manipulating the subsets of training instances perform especially well for unstable learning algorithms, such as decision tree, neural networks, or rule-based learning algorithms (Dietterich, 1997). Variability of their predictions can be reduced due to the training set variability. Their output classifier undergoes major changes in response to small changes in the training data. Nearest Neighbor, linear regression, and linear threshold algorithms are stable when the training sample size is reasonably large and unstable when it is quite small (Skurichina, 2001).

As proposed in Breiman (2000), consider the training set as consisting of M independent draws from the same underlying distribution. Conceptually, training sets of size M can be drawn repeatedly, and the same learning algorithm will be used to construct a predictive model. Those models will vary, and the extent of this variability is a dominant factor in producing generalization error. The way to reduce the variability realized in sampling techniques is perturbation of the training instances to produce alternative training sets (samples), and constructing multiple predictive models to be integrated, for example, by voting.

Bagging (Breiman, 1996) is based on a repeated bootstrap sampling (Efron & Tibshirani, 1993) and an aggregation procedure. Bootstrap sampling uses sampling with replacement, thus in a sample some instances appear more than once, and some instances are not included at all. It has been shown that on average 63.2% of the instances of the learning set are included in the sample at least once (Breiman, 1996; Bauer & Kohavi, 1999). Then a classifier is constructed on this sample by a learning algorithm. Generation of new samples continues until the desired number of component classifiers is achieved. Bootstrapping helps to avoid or get less non-representative instances in the training set.

In order to generate a component classifier a random sampling is performed drawing a sample of M instances having repeated instances. Then a classifier is constructed on this sample by a learning algorithm. Generation of new samples continues until the desired number of component classifies is achieved. Bootstrapping helps to avoid or get less non-representative instances in the training set.

Boosting is generally proven as more accurate than bagging, although performance of boosting is more variable than performance of bagging (Opitz & Maclin, 1999; Tumer & Ghosh, 1996b). Boosting is the most efficient for large

training sample sizes while bagging and RSM are beneficial for small and critical sample sizes (Skurichina, 2001).

Boosting scheme, illustrated by the AdaBoost algorithm (Freund & Schapire, 1996; Freund & Schapire, 1997), also chooses a training set of size $M$and initially sets the probability of picking each instance to be $1/M$. At each iteration, these probabilities are changed according to error rate of each classifier $h_s, s = 1 \ldots S$, weighted with the probabilities of instances incorrectly classified by h_s. The component classifiers are then integrated using weighted voting. The effect of the change in weights is to place more weight on training instances that were misclassified by h_s, and less weight on instances that were classified correctly.

AdaBoost algorithm requires weak classifiers produced, for example, by decision trees or neural networks to be integrated. Their error rate should be bounded by a constant strictly less than 0.5. Schapire *et al*. (1998) provided an explanation for the fact that the generalization error does not increase when many classifiers are combined.

Another sampling technique called cross-validation partitioning (Parmanto *et al.*, 1996) is based on the procedure employed in the cross-validation majority technique. Samples are constructed leaving out disjoint subsets of the training set. Training set is randomly divided onto l folders and l overlapping samples are generated each time dropping out one of the folders.

2.2.3 Combined ensemble techniques

The approach proposed in this thesis for heterogeneous classification problems is a synergy of multiple feature subspace projection, sampling, and class encoding. In this subsection related works on ensemble techniques combining multiple feature subspace projection, sampling, and class encoding ensemble generation methods are considered. Combined ensemble techniques derive benefits in some situations uniting the strengths of different techniques and avoiding their weaknesses. The advantage is demonstrated as the measured ensemble performance characteristics, that is training/generalization accuracy, diversity, complexity, and error bias/variance reduction.

Several recent research works studied combination of ensemble techniques for various learning algorithms. Mostly, different sampling techniques have been combined with boosting, which is effective for a wide variety of classification tasks. However, only a few studies have been done on combination of multiple feature subspace projection with class encoding and multiple feature subspace projection with sampling techniques, which can bring potential advantage for heterogeneous classification problems.

For example, in Oza and Tumer (1999) a combination of correlation-based feature subset selection and one-per-class decomposition has been proposed to generate the component classifiers. This method called Input Decimation is based on the assumption that features highly correlated with particular class and uncorrelated between each other are important to classify instances from

that particular class. This reduces correlation between errors produced by the component classifiers promoting ensemble diversity and accuracy growth.

In Fürnkranz (2002) a straightforward combination of pairwise ensemble technique and bagging has been explored for a decision tree learning algorithm. A number of classifiers in ensemble has been increased by 10 times since for each pairwise partitioning 10 samples with replacement were drawn. The obtained component classifiers were integrated through simple voting. As a result, performance of the pairwise technique was considerably improved.

Considerable increase of classification accuracy for the Nearest Neighbor learning algorithm is obtained by combining ECOC with local feature selection proposed in Ricci and Aha (1997a).

Guruswami and Sahami (1999) extended ECOC performance in solving multiclass problems with the power of boosting to annul the error correlation disadvantages of ECOC. Feature selection was employed for each output coding that significantly increased classification accuracy on several multi-class data sets, and the conditions under which the method works were explained.

In Windeatt and Ardeshir (2002) combination of boosting and ECOC is explored for decision trees. Experiments have shown that this technique demonstrates better accuracy for unpruned decision trees.

In Lazarevič *et al*. (2000) an Adaptive Attribute Boosting technique has been proposed to coalesce multiple local classifiers, which are constructed at each boosting round on different subsets of features. The benefits of this technique have been demonstrated on heterogeneous spatial data sets.

Several studies have been performed on combination of class encoding with boosting and/or bagging, and multiple feature subspace projection with boosting and/or bagging that enhances the effect of formers. For example, Zheng, Webb and Ting (1998) combined boosting with stochastic attribute selection and showed that the combined technique effectively increases ensemble diversity and accuracy. In Zheng and Webb (1998) authors enhanced this combination incorporating bagging and got further increase of accuracy and reduction of variability. Combination of multiple feature subspace projection and sampling techniques was also explored by Breiman (2001). In that paper, random feature selection is enhanced with bagging.

In Skrypnyk and Ho (2003) it was indicated that the mechanisms of accuracy increase offered in bagging and boosting in combination with random feature selection are different from those required for heterogeneous classification problems. Such sampling techniques as bagging and boosting pick up nearly equal number of instances representing different subproblems in synthetic data. Though, combination of boosting and feature selection based one-per-class ensembles promotes accuracy growth for some data sets, as shown in Skrypnyk *et al*. (2003). It can potentially help to improve accuracy in subproblems associated with homogeneous regions.

2.3 Chapter summary

In this chapter, the classification and clustering tasks are introduced under a unified framework. The related notion of feature relevance in classification and the feature selection task are considered. These tasks are subsidiary to multi-model approach in classification, which is introduced thereafter. Basic notions are described from perspectives of machine learning and statistical decision theory. Introduction of theoretical and structural basis of ensemble learning is followed by categorization of ensemble techniques and brief overview of the relevant findings. This chapter provides the essential backgrounds for understanding the problem of unstable feature relevance in classification, the concept of heterogeneous classification problems, and the developed solutions based on heterogeneity decomposition.

Classification is a prediction of the class labels for the structured domain examples (instances) using a model build on the similar instances with known class labels. The C4.5 decision tree, Naïve Bayes, and *k*-Nearest Neighbor that exemplify different approaches to learning are described in Appendix 2. A particular attention is given to their performance in high dimensions response to the presence of irrelevant features. Clustering task deals with discovery of new classes and structural relationships between categories in data. In this thesis it has been considered with respect to finding homogeneous regions in classification problems. Basic clustering techniques are briefly introduced in Appendix 2. Selection of relevant features addressed in this chapter is considered with respect to classification task. The aim of feature selection is to find a subset of relevant features to increase predictive accuracy and/or simplify a learning model. The notions of relevant, redundant and interactive features are considered. A BDP multi-model approach follows the divide-and-conquer paradigm in data mining. Ensembles learning based on multiple models provides a convenient tool for decomposition and mechanisms for prediction accuracy improvement.

3 HETEROGENEOUS CLASSIFICATION PROBLEMS AND DECOMPOSITION APPROACHES

This chapter addresses the problem of unstable feature relevance in predictive data mining introducing the concept of heterogeneous classification problems. A few basic heterogeneity types are considered in Section 3.1. Decomposition as a generalized solution is suggested in Section 3.2. Under ensemble framework, decomposition is a part of the learning stage followed by construction of local models covering homogeneous regions. The chapter provides an introduction for two approaches to perform decomposition of a heterogeneous classification problem, at the class level and beyond the class level, which are presented in Chapters 4 and 5.

Decomposition approaches based on local feature relevance evaluation and local class separability estimation considered in subsections 3.2.2 and 3.2.3 are conceptually similar, but follow different decomposition schemes. The first one is applied at the class level, and the second one reaches beyond class labels. The search strategies used to find candidate local regions that possess homogeneity and related evaluation criteria are briefly discussed. Evaluation functions and search strategies specific to both approaches are elaborated in the subsequent chapters. Section 3.3 presents decomposition within an ensemble framework and describes ensemble generation using a general decomposition scheme and integration of the component classifiers.

3.1 The classification heterogeneity

During the last decade, the problem of heterogeneity in data have been addressed in several machine learning studies under different names, such as local feature relevance, attribute instability and relevance in context. This problem has been forestalled in earlier machine learning research on context-sensitive and local learning, for example, discussing relevance of features in context (Domingos, 1997) and local feature relevance (Howe & Cardie, 1997; Friedman, 1994). The problem of heterogeneity in contemporary data has been introduced recently as a problem related to classification tasks (Apte *et al*., 1998; Lazarevič *et al*., 2000).

At the same time in the databases research community integration of heterogeneous databases has become an actively discussed topic (Dey *et al*., 2002). As a result, many issues related to mining massive heterogeneous databases were brought on top, including classification tasks (Pinheiro & Sun, 1998, Fuseida & Satou, 1999). The data sets obtained from heterogeneous databases are often used for prediction in different areas, for example, in bioinformatics (Pavlidis *et al*., 2001, Thierry-Mieg, 2000) and medicine (Golub *et a*l., 1999; Ramaswamy *et al*., 2001).

In this thesis, unstable feature relevance is considered as a part of a bigger problem, a problem of classification heterogeneity. A starting point for this research became a paper by Apte *et al*. (1998), where classification heterogeneity of two types is described, feature space heterogeneity and class heterogeneity. This thesis uses basic terminology established in this paper. The extended definitions of the heterogeneity types used throughout the manuscript are presented later in this section. Before formalizing the definition of classification heterogeneity, early research works discussing this phenomenon and pointing out to the problem will be overviewed.

3.1.1 Unstable feature relevance: early works on local feature selection

The fact that feature relevance may vary across the set of instances was elaborated in early works on feature weighting for lazy learning algorithms, mostly for the *k*-Nearest Neighbor learning algorithm. Two representative local weighting schemes for *k*-Nearest Neighbor (Hastie & Tibshirani, 1996; Friedman, 1994) are instance-specific. In Hastie and Tibshirani (1996) a separate distance metric for each target instance is computed through an iterative process. In Friedman (1994) the most relevant feature is scaled at each step such that a fixed fraction of the given training instances fall outside of a predetermined range around the target instance. The training instances outside of that range are then discarded, the new most relevant feature is determined, and the process is repeated until only k training instances remain. The local relevance of each feature is estimated from the estimated reduction in classification error. Both papers report favorable results on their local approaches comparatively to global ones, but both of those local approaches

proposed are computationally expensive.

Howe and Cardie (1997) have proposed a coarsely local feature weighting scheme, class distribution weighting, where feature weights are allowed to vary being identical for certain clusters of instances. This method is a precursor for several class encoding feature selection and ensemble techniques, for example proposed in Oza and Tumer (1999) and Hall (1999), which assume that there are features that are useful at distinguishing whether an instance is of one particular class, but are not useful at distinguishing between the remaining classes. The method presented in Howe and Cardie (1997) is based on the assumption that *although classes are not always homogeneous, it is plausible that for many domains features informative for a particular class are the same for most or all instances belonging to that class*.

Domingos (1997) has extended the concept of local feature relevance. He motivated that *some features may be highly relevant in certain regions of the instance set being irrelevant everywhere else by their sensitivity to a context that is to the values of the other features*. The Relevance-in-Context method proposed is distance-based and instance-specific that makes it computationally expensive. In Domingos (1997) a *feature difference* measure is also considered to evaluate the context dependency effect exhibited by Relevance-in-Context in real data sets.

Alternatively, the idea of contextual features has been explored by Turney (1996, 1993). He distinguishes three different types of features: *primary*, *contextual*, and *irrelevant* features. By his definition, primary features are useful for classification when considered in isolation, without regard for the other features. Contextual features are not useful in isolation, but can be useful when combined with other features. Irrelevant features are not useful, either when considered alone or when combined with other features. In those works, different strategies for exploiting context have been studied, assuming that contextual features are available for learning.

Harries and Horn (1996) recognize hidden changes in context for concept learning and propose a batch learning approach. They identify *environmental* features, which reflect hidden context. The examples of environmental features are time or spatial location. The SPLICE method proposed is a meta-level algorithm that uses a learning algorithm capable to perform context-sensitive feature selection, like Relevance-in-Context, or decision tree. Such learning algorithms select features at each node, rule, or clause in the context of locally relevant prior selections. Partitions made by them over some environmental feature selected (for example, time feature) identify possible changes in context. Then contextual clustering is performed over the intervals according to apparent similarity of context, and local context-specific concepts are learned. SPLICE does not seek or extract the contextual features from heterogeneous data, assuming that context may be contiguous over some ordinal features (environmental).

In Apte *et al*. (1998) a method to search for evidence of heterogeneity and a method to decompose the problem into constituent subproblems are proposed. A transformed cosine similarity measure, the Importance Profile Angle (IPA),

has been suggested to perform a test for heterogeneity. IPA reflects the degree of dissimilarity between the decision boundaries for a pair of subproblems. In IPA profiles of feature importance are compared based on some feature merit measure used to create a scored rank. Authors in Apte *et al*., (1998) have developed a contextual merit measure and compared it to Information gain and ReliefF measures. If any heterogeneity is found, a tree–like search procedure is used to perform splits of the data set until homogeneous regions are found. This strategy is applicable in situation when contextual features are available, for example, age or gender. Then a spit is performed by the values of those contextual features, implying that predictive models will be different for males and females, or for children and adults.

Authors in Apte *et al*. (1998) have shown that heterogeneity may appear at the class level: a set of features to discriminate one class from the other class(es) is different from a set of informative features associated with another class. A similar line of thinking can be observed also in Cardie and Howe (1997), Hall (1999), and Oza and Tumer (1999). Authors in Apte *et al*. (1998 suggested that differences in class probability distribution may give an indirect indication for decomposition. Therefore, they based their heterogeneity test solely on feature relevance profiles in subproblems.

Dependence between contextual and primary features as a particular case of feature dependencies has been considered in Robnik-Šikonja and Kononenko (1996) and Robnik-Šikonja and Kononenko (1999). Robnik-Šikonja and Kononenko (1996) state that Relief and its extension ReliefF are both capable to estimate correctly the quality of features in classification problems with strong feature dependencies. By exploiting the local information provided by different contexts they provide a global view and recognize contextual features.

In the original Relief (Kira & Rendell, 1992) the quality of features is estimated according to how well their values distinguish between the instances that are near to each other, evaluating two nearest neighbors of the target instance - from the same and different class. When calculating a merit measure for a feature Relief, similar to contextual merit, takes into account correlations between features contrary to the "myopic" feature merit measures that assume feature independence, such as Information gain and Gini index.

Later works dealing with heterogeneous data and unstable feature relevance use ensemble learning. Class encoding ensemble techniques considered in the next section originally have been designed to solve multi-class classification problems. A new use of class encoding has been proposed in Oza and Tumer (1999). The Input Decimation method described in this paper generates one component classifier per each class using a subset of features correlated with this class.

In Lazarevič *et al.* (2000) the Adaptive Boosting technique for heterogeneous spatial databases with unstable feature relevance has been developed. Authors point out that in heterogeneous databases there are features that change their relevance across the instance set. In the boosting round of Adaptive Boosting the local information for the drawn sample is

maximized by feature selection, extraction or weighting and at the same time, spatial data blocks are drawn.

These are a few examples of unstable feature relevance considered in the literature a decade ago. From that literature we have observed two tendencies in solving the unstable feature relevance problem: (1) data structure oriented or class separability oriented, and (2) identification of contextual features or increasing sensitivity to context changes. Currently, the problem is recognized mostly with respect to different application domains. This thesis considers examples from medical and biomedical domains in the experimental section.

The decomposition approaches and heterogeneity types proposed in this thesis are conforming to these tendencies, with an attempt to extend, formalize, categorize, and unify them all.

3.1.2 Classification heterogeneity types

In many prediction tasks, characteristics of the class boundaries are very different in different regions of the feature space (Ho & Basu, 2002; Pierson, 1998). The spatial location of those homogeneous regions has a vital importance. In this situation often the contribution of features to discriminate between classes may be unequal across the set of instances. Thus, in general, classification heterogeneity is *feature space heterogeneity*.

Data structure differs in homogeneous regions having different relevant features. It may happen that in a particular feature subspace some group of instances can be separated from the others with a simple boundary (for example, linear or piecewise-linear boundary). In addition, this group may contain instances of a particular class, or a subset of classes. It means that those features are relevant to discriminate classes at this particular group of instances while being less useful or completely useless for the rest of instances.

A particular case of classification heterogeneity when the homogeneous regions are composed by instances of one particular class, or a subset of classes, is denoted in Apte *et al*. (1998) as *class heterogeneity*. In this case, the decision rules that distinguish one subset of classes from another might be different from those that discriminate classes within this subset.

Class heterogeneity occurs in practical prediction tasks quite often, because the data sets usually have some structure resulting from non-random processes that generated the data. For classification tasks, this structure is related to class labeling and grouping of instances. Thus, homogeneous regions may often correspond to class labeling (Skrypnyk, 2004).

Interactions, in particular, higher order dependencies between features are important characteristics of heterogeneous classification problems. Sometimes local relevance of other features, and hence, grouping of instances at homogeneous regions is specified by the values of so-called *contextual features.* It can be illustrated by the following example from medical domain considered in Apte *et al*. (1998). In medical diagnostics diagnosis may require quite different predictive models for different genders, thus a feature "gender" by its values

specifies different symptoms (relevant features) to consider for females and males.

Sometimes grouping of instances at homogeneous regions can be clearly seen in a particular projection of a feature space to a restricted subspace. Important roles in this situation play contextual features. By their values, they specify groups of instances and locally relevant features.

In order to demonstrate how projections of those groups in corresponding feature subspaces improve class separability, consider an example on continuous data illustrated in Figures 3 – 6. This synthetic data set has 7 features following Gaussian distributions; feature f_0 by two intervals of its values specifies local relevance of feature subsets (f_1, f_2, and f_3) and (f_4, f_5, and f_6) in homogeneous regions. There are two homogeneous regions each containing instances of both classes.

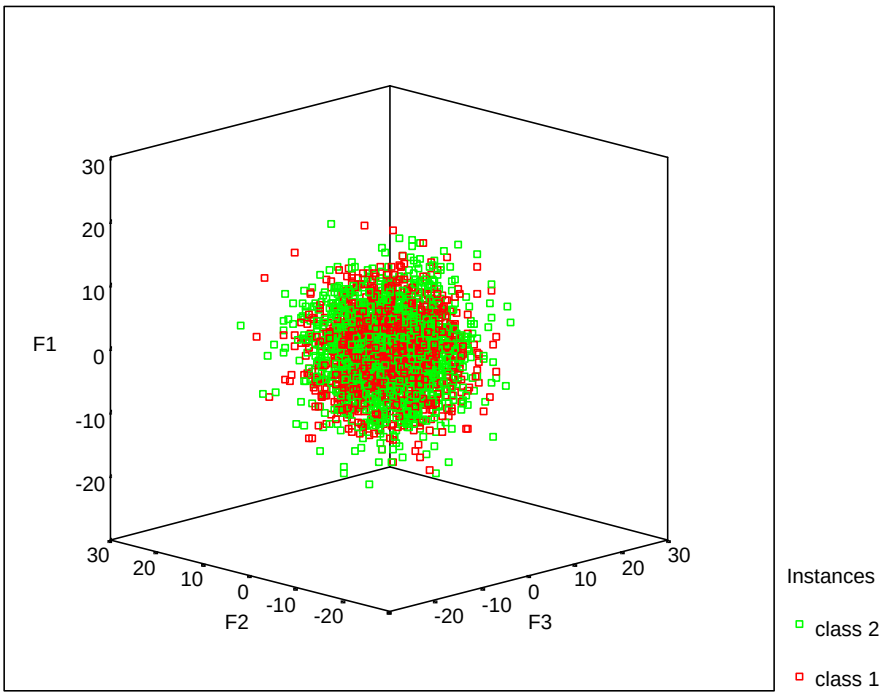

FIGURE 3 7-dimesional Gaussian data shown in the subspace of features $f_1, f_2,$ and f_3. Instances belonging to different classes are of different colors. In these dimensions class boundaries are unseen.

One may see that in some arbitrary projection of the feature space classes (and class boundaries) heavily overlap, as shown in Figure 3. When contextual feature(s) are used in projections distribution of instances from different classes is different along the corresponding dimensions (Figure 4). After decomposition of the classification problem into two subproblems, where homogeneous regions are modeled separately, classes become easier separable (Figures 5 and 6).

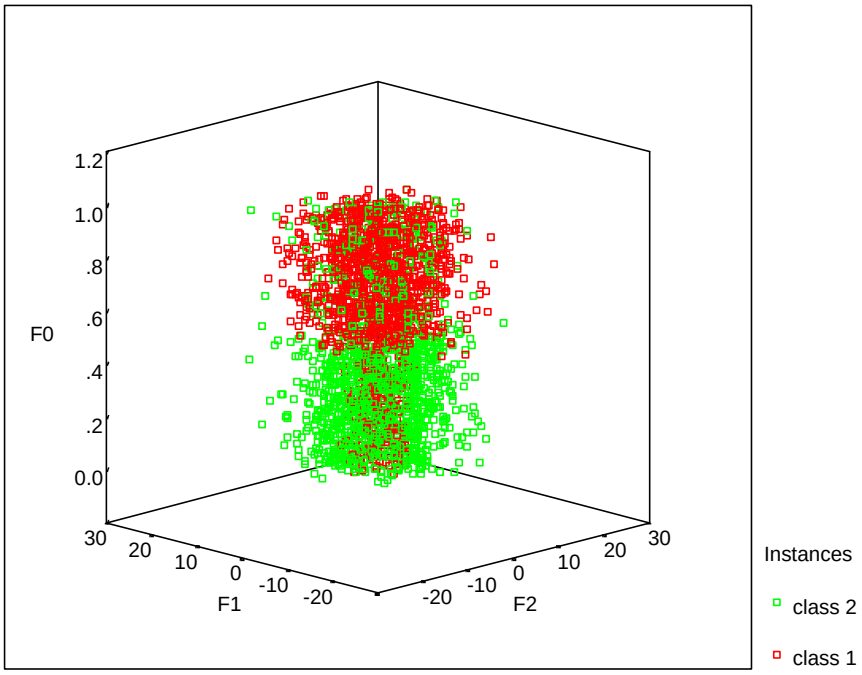

FIGURE 4 7-dimesional Gaussian data shown in the subspace of features $f_0, f_1,$ and f_2. Feature f_0 is contextual. Distribution of instances from different classes is different along this dimension.

Often the domain knowledge is limited and contextual features are, which by their values can specify some division onto homogeneous regions, are not present or unknown. Consider an example of feature space heterogeneity where homogeneous regions are not related to a particular class labeling.

A data set contains all legal 8-ply positions in the Connect-4 6 x 7 board game (Allen, 1990) in which neither player has won yet and the next move is not forced. From these positions, it is necessary to predict either win/loss for the first player, or draw. In this data set each cell is represented by a feature taking three possible values: x – the first player has taken, o – the second player has taken, and b – blank. In order to make a prediction for each case (instance) there is no need to consider all cells (features). The features relevant for prediction may change from case to case. Obviously, the subset of features taking x or o values in the particular instance is always relevant for this instance.

This example illustrates an extreme case when each instance in a data set has a unique subset of relevant features. In general, instances can be grouped according to the relevance degree of different features, or by subsets of relevant features. Considering such grouping one can assume the following situations.

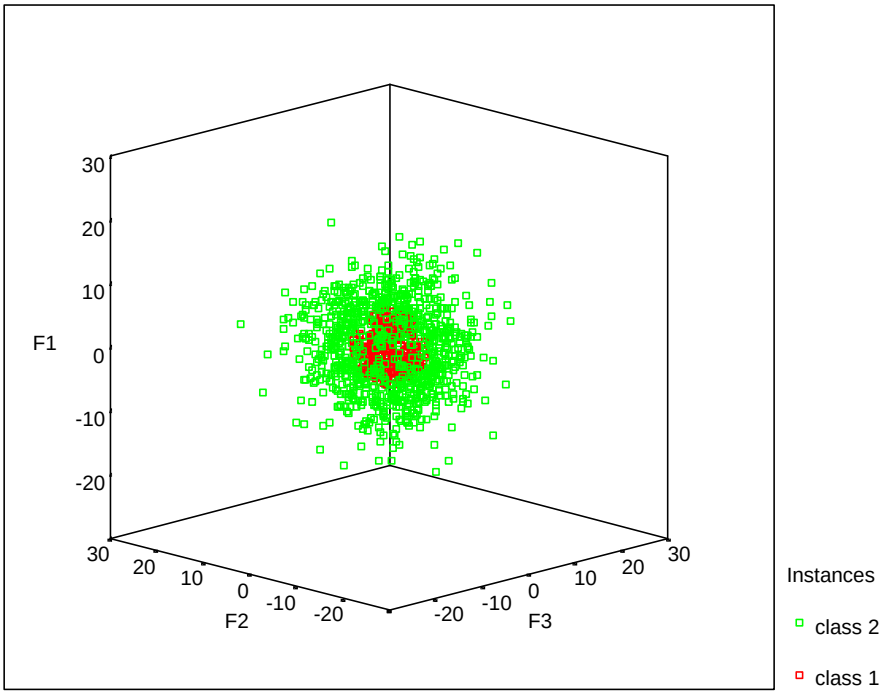

FIGURE 5 7-dimesional Gaussian data shown in the subspace of features $f_1, f_2,$ and f_3 after decomposition. Data set is decomposed on two subproblems according to the particular values of the contextual feature f_0. In these dimensions it is visible that instances from class 2 surround instances from class 1.

Subsets of relevant features for different groups may be disjoint or a subset at one region may have a superset at another region. Division by groups assumes that each group includes sufficient number of instances to describe a homogeneous region and to construct a local model using those instances.

There are variations of feature space and class heterogeneity. In Table 1 a general case of heterogeneity is shown schematically.

TABLE 1 An interpretation of a general case of heterogeneity in data. Relevant features are marked by **R**, and irrelevant ones by I. Each group includes all the instances for which the combination of relevant and irrelevant features holds.

Instances	**Features**							**Class**
x	f_1	f_2	f_3	…	f_j	…	f_N	y
Group 1	**R**	**R**	*I*	…	**R**	…	**R**	$c_1, \ldots, c_D$
Group 2	I	**R**	**R**	…	**R**	…	**R**	$c_1, \ldots, c_D$
Group 3	I	I	**R**	…	I	…	**R**	$c_1, \ldots, c_D$
…	…	…	…	…	…	…	…	…
Group i	**R**	I	I	…	I	…	**R**	$c_1, \ldots, c_D$
…	…	…	…	…	…	…	…	…
Group Z	I	**R**	**R**	…	I	…	**R**	$c_1, \ldots, c_D$

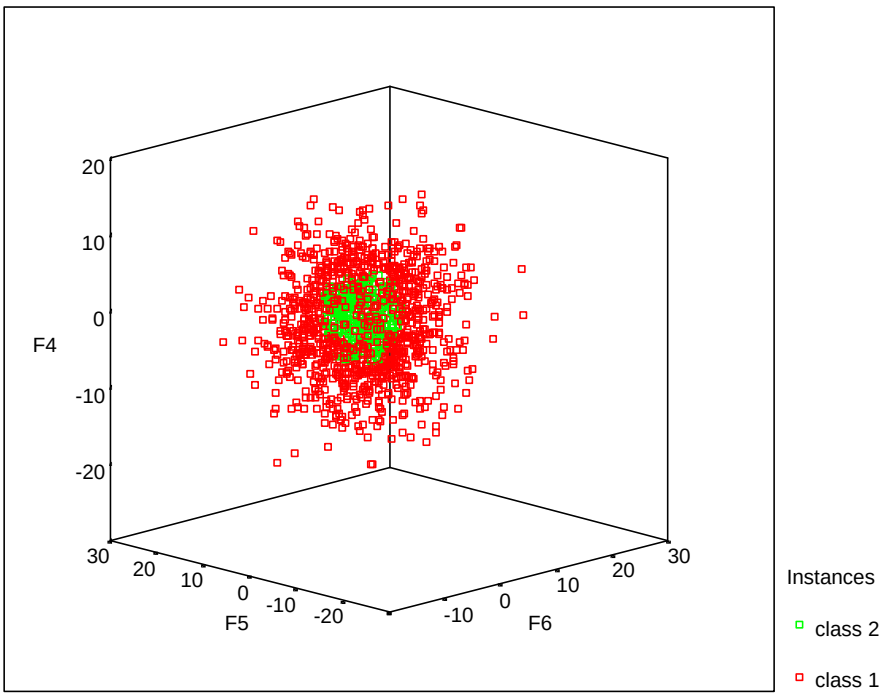

FIGURE 6 7-dimesional Gaussian data in the subspace of features f_4, f_5, and f_6 after decomposition. Data set is decomposed on two subproblems according to the particular values of the contextual feature f_0. In these dimensions it is visible that instances from class 1 surround instances from class 2.

Instances are grouped according to coincidence in their relevant features. Globally relevant features (such as feature f_N in Table 1) appear in every group. Each group may include instances from different classes.

Each group may be composed of instances that belong to different classes, which means that heterogeneity is not related to class labeling. In practice, different groups may include instances from the particular subsets of classes. This is a particular case of heterogeneity, class heterogeneity schematically presented in Table 2.

In the particular case of class heterogeneity, *one-class heterogeneity*, different subsets of features are considered relevant to distinguish one class from the others. This case is shown schematically in Table 3.

Sometimes the domain knowledge regarding heterogeneity is available. For example, in medical diagnostic problems for different individuals various clinical analyses, tests, symptoms, and anamneses recorded as features may be needed to identify same diseases. Then individual's sex, age group, belonging to some risk category by individual's employment, residence, and so on, are contextual features.

It is assumed that a contextual feature by its values may specify groups of instances with identical subsets of relevant features. Those relevant features, specific for each group, are called *primary* features. In case of one-class

heterogeneity, when contextual features specify groups inside which all instances belong to the same class, sometimes there is no need to consider primary features.

TABLE 2 An interpretation of class heterogeneity. Classes within subsets are separable in a particular set of dimensions (features) specific for this subset of classes. Relevant features are marked by **R**, and irrelevant ones by I. Each group includes all the instances for which the combination of relevant and irrelevant features holds.

Instances	**Features**							**Class**
x	f_1	f_2	f_3	…	f_j	…	f_N	y
Group 1	**R**	**R**	*I*	…	**R**	…	**R**	$c_i, c_j, \dots, c_k$
Group 2	I	**R**	**R**	…	**R**	…	**R**	$c_l, c_m, \dots, c_n$
Group 3	I	I	**R**	…	I	…	**R**	$c_p, c_r, \dots, c_s$
…	…	…	…	…	…	…	…	…
Group i	**R**	I	I	…	I	…	**R**	$c_t, c_u, \dots, c_v$
…	…	…	…	…	…	…	…	…
Group Z	I	**R**	**R**	…	I	…	**R**	$c_g, c_q, \dots, c_w$

TABLE 3 An interpretation of one-class heterogeneity. Each class is separable from the rest of classes in a particular set of dimensions (features) specific for this class. Relevant features are marked by **R**, and irrelevant ones by I. Each group includes all the instances for which the combination of relevant and irrelevant features holds. Groups include different number of instances.

Instances	**Features**							**Class**
x	f_1	f_2	f_3	…	f_j	…	f_N	y
Group 1	**R**	**R**	*I*	…	**R**	…	**R**	c_1
Group 2	I	**R**	**R**	…	**R**	…	**R**	c_2
Group 3	I	I	**R**	…	I	…	**R**	c_3
…	…	…	…	…	…	…	…	…
Group i	**R**	I	I	…	I	…	**R**	c_d
…	…	…	…	…	…	…	…	…
Group Z	I	**R**	**R**	…	I	…	**R**	c_D

Contextual features may specify relevance of primary features by their values. Several contextual features may be interacting, that is they specify relevance of primary features and group them by combination of their values. This case is schematically shown in Table 4.

It is not necessary that each of all possible combinations of the contextual features values defines a group. At the same time some features may be locally relevant, but not under control of contextual features (the feature f_j in Table 4). Some features may be globally relevant (the feature f_N in Table 4) or relevant for several groups. Similar to examples provided in Table 2 and Table 3, each group defined by contextual features may include instances from different subsets of classes (class heterogeneity) or an individual class (one-class heterogeneity). Class heterogeneity defined by P contextual features is shown in

Table 5. In this case, contextual features by their particular values combination specify relevance of the primary features, which unambiguously identify subsets of classes.

TABLE 4 Contextual heterogeneity with *P* features. Relevant features are marked by **R**, and irrelevant ones by I. Each group includes all the instances for which the combination of relevant and irrelevant features holds. Groups include various numbers of instances. Relevance of the feature f_j does not depend on contextual features. The feature f_N is globally relevant.

Instances	**Contextual features**			**Primary features**							**Class**
x	c_1	…	c_P	f_1	f_2	f_3	…	f	…	f_N	y
	$v_{1,i}$	…	$v_{P,r}$	**R**	**R**	I	…	**R**	…	**R**	
	…	…	…	**R**	**R**	I	…	**R**	…	**R**	$c_1, \ldots, c_D$
	$v_{1,j}$	…	$v_{P,s}$	**R**	**R**	I	…	I	…	**R**	
…	…	…	…	…	…	…	…	…	…	…	…
	$v_{1,k}$	…	$v_{P,t}$	I	I	**R**	…	**R**	…	**R**	
	…	…	…	I	I	**R**	…	I	…	**R**	$c_1, \ldots, c_D$
	$v_{1,l}$	…	$v_{P,q}$	I	I	**R**	…	**R**	…	**R**	
…	…	…	…	…	…	…	…	…	…	…	…
	$v_{1,m}$	…	$v_{P,g}$	**R**	I	**R**	…	I	…	**R**	
	…	…	…	**R**	I	**R**	…	I	…	**R**	$c_1, \ldots, c_D$
	$v_{1,n}$	…	$v_{P,w}$	**R**	I	**R**	…	**R**	…	**R**	

TABLE 5 Class heterogeneity with *P* contextual features. Relevant features are marked by **R**, and irrelevant ones by I. Each group includes all the instances for which the combination of relevant and irrelevant features holds. Groups include various numbers of instances. Relevance of the feature f_j does not depend on contextual features. The feature f_N is globally relevant.

Instances	**Contextual features**			**Primary features**							**Class**
x	c_1	…	c_P	f_1	f_2	f_3	…	f_j	…	f_N	y
	$v_{1,i}$	…	$v_{P,r}$	**R**	**R**	I	…	**R**	…	**R**	
	…	…	…	**R**	**R**	I	…	**R**	…	**R**	$c_i, c_j, \ldots, c_k$
	$v_{1,j}$	…	$v_{P,s}$	**R**	**R**	I	…	I	…	**R**	
…	…	…	…	…	…	…	…	…	…	…	…
	$v_{1,k}$	…	$v_{P,t}$	I	I	**R**	…	**R**	…	**R**	
	…	…	…	I	I	**R**	…	I	…	**R**	$c_t, c_u, \ldots, c_V$
	$v_{1,l}$	…	$v_{P,q}$	I	I	**R**	…	**R**	…	**R**	
…	…	…	…	…	…	…	…	…	…	…	…
	$v_{1,m}$	…	$v_{P,g}$	**R**	I	**R**	…	I	…	**R**	
	…	…	…	**R**	I	**R**	…	I	…	**R**	$c_g, c_q, \ldots, c_w$
	$v_{1,n}$	…	$v_{P,w}$	**R**	I	**R**	…	**R**	…	**R**	

One-class heterogeneity defined by *P* contextual features is shown in Table 6. In this case, contextual features by their particular values combination specify

relevance of the primary features, which unambiguously identify instances from a particular class. This makes primary features redundant.

TABLE 6 One-class heterogeneity with P contextual features. Relevant features are marked by **R**, and irrelevant ones by I. Each group includes all the instances for which the combination of relevant and irrelevant features holds. Groups include various numbers of instances. Relevance of the feature f_j does not depend on contextual features. The feature f_N is globally relevant.

Instances	**Contextual features**			**Primary features**							**Class**
x	c_1	…	c_P	f_1	f_2	f_3	…	f_j	…	f_N	y
	$v_{1,i}$	…	$v_{P,r}$	**R**	**R**	I	…	**R**	…	**R**	c_1
	…	…	…	**R**	**R**	I	…	**R**	…	**R**	
	$v_{1,j}$	…	$v_{P,s}$	**R**	**R**	I	…	I	…	**R**	
…	…	…	…	…	…	…	…	…	…	…	…
	$v_{1,k}$	…	$v_{P,t}$	I	I	**R**	…	**R**	…	**R**	c_d
	…	…	…	I	I	**R**	…	I	…	**R**	
	$v_{1,l}$	…	$v_{P,q}$	I	I	**R**	…	**R**	…	**R**	
…	…	…	…	…	…	…	…	…	…	…	…
	$v_{1,m}$	…	$v_{P,g}$	**R**	I	**R**	…	I	…	**R**	c_D
	…	…	…	**R**	I	**R**	…	I	…	**R**	
	$v_{1,n}$	…	$v_{P,w}$	**R**	I	**R**	…	**R**	…	**R**	

Feature relevance in heterogeneous classification problems characterizes ability of features to discriminate between classes in a particular group of instances. Relevance of features in local regions of the instance set is specified by a context, for example, by the values of contextual features or by the neighboring instances in terms of some distance metric, especially when contextual features are not present.

3.2 Decomposition approaches

In this section author suggests different decomposition approaches for different types of heterogeneity. Two main constituents of these approaches are search and evaluation. Search strategies and evaluation criteria depend on a heterogeneity type.

3.2.1 Classification heterogeneity decomposition basics

The approaches to construct predictive models on heterogeneous data currently presented in the literature are mostly based on accommodation of the prior domain knowledge. Mainly, those approaches were developed for a particular application task in collaboration with the domain experts (Golub *et a*l., 1999; Ramaswamy *et al*., 2001; Pavlidis *et al*., 2001; Thierry-Mieg, 2000). When the origins of heterogeneity are unknown and the domain knowledge is limited, construction of local models becomes challenging and requires approaches based on assumptions about the problem domain.

There are two main directions toward developing an approach to construct predictive models on heterogeneous data with limited domain knowledge. The first one is to perform decomposition of a heterogeneous classification problem with local models covering homogeneous regions in the data set, one or more models per region, according to the divide-and-conquer principle. Similar approach has been applied, for example, in Lazarevič and Obradovič (2001a) for spatial agricultural data. The second one is to apply a local similarity based learning procedure that is capable to handle instability of features' relevance, also known as lazy learning, or local learning. Local learning methods do not provide a complete description of the input/output mapping but rather approximate the function in the neighborhood of the instances to be predicted. However, this approach is very intensive computationally and mainly associated with the necessity of a possibly large amount of memory to store the data.

A decomposition approach divides a classification problem into simpler subproblems. Solutions to subproblems in combination yield a solution to the original problem. This approach has two main advantages over the global modeling approach. First, the choice of the local model complexity and the estimation of parameters can rely on the linear techniques, well studied and developed over the years. Second, it provides better adjustment to the properties of the data set. Mathematically, the problem of function estimation turns to the problem of value estimation. (Bontempi & Birattari, 2005)

There are 4 steps that should be performed following the decomposition approach: (1) location of homogeneous regions has to be approximated finding corresponding groups of instances in feature subspaces, (2) local regions have to be modeled individually, (3) the group membership of a new unclassified

instance has to be identified according to a certain rule, for example, description of the approximated regions, and (4) the local model(s) associated with a homogeneous region has to be applied to predict class of the new instance.

At the first step, a search procedure and an evaluation function have to be chosen in order to find and evaluate the candidate local regions as groups of instances. Each group of instances has to be modeled using a subset or relevant features that also have to be found. Simultaneous grouping of instances in feature subspaces is an optimization task, where the criterion is specified by the evaluation function.

The two decomposition approaches proposed in Subsection 3.2.2, 3.2.3, and 3.2.4 accordingly use different evaluation functions and search strategies. In this chapter, we mainly focus on search strategies. Steps (2)-(4) are presented within an ensemble learning framework in Section 3.3. Evaluation of candidate homogeneous regions and all related measures are detailed in Chapters 4.

3.2.2 Decomposition based on local feature relevance evaluation

Contribution of different features to discriminate between classes is different and varies across the set of instances when the data is heterogeneous. This is a general definition of heterogeneity and implies the following problem statement: how to find / approximate homogeneous regions in order to evaluate local feature relevance? The natural approach is to examine different groups of instances estimating local relevance of features inside the groups and use some evaluation function to find the best decomposition. After decomposition of the instance set into groups according to the approximated homogeneous regions, a subset of relevant features, unique for each group, will be used to build a local model to cover that region. A subset of locally relevant features can be found using feature selection methods over the corresponding subset of instances.

The most problematic part of this approach is to find the candidate local regions, in other words, groups of instances. The simplest way is to use random sampling with replacement to find the candidate instance subsets and then apply an evaluation function and a weighting scheme for instances with respect to groups. Evaluation function may be based on the difference in the features importance locally and globally as measured in terms of some feature ranking or feature weighting method.

One-class heterogeneity assumption (Table 3) makes decomposition of the data set straightforward. The local regions consist of instances belonging to the same class. In this case one-against-all or one-against-one class encoding can be performed (Valentini & Masulli, 2002). Ensembles based on one-per-class and pairwise class decompositions are able to provide accuracy improvement for classification tasks that are not heterogeneous (Masulli & Valentini, 2000)

The search for local regions can be guided by the contextual features if they are available. Then, the first step is to identify the contextual features. If they are determined, decomposition of the data set into homogeneous regions

can be performed by the values of the contextual features. After that, locally relevant subsets of features can be found.

One-class heterogeneity with contextual features (Table 6) reduces to a homogeneous classification problem. In this case, no difference whether the contextual features are independent or not, the set of contextual features is enough for the perfect class discrimination. There can be a higher order dependency between the contextual features (Chapter 4), when neither is individually predictive of the class, but they are relevant together, in a subset. Feature selection algorithms that are based on individual feature evaluation cannot identify the contextual features correctly in this case. The feature ranking algorithms will assign low rank to interacting features as consider the primary features as partially predictive. The synthetic example of this situation will be considered in the experimental section.

In the general case of class heterogeneity contextual features by their values may suggest decomposition of the data set into group each represented by subsets of classes (Table 5). It means that discrimination within a certain subset of classes needs a different decision rule, hence a different model, comparatively to another subset of classes. A known class encoding scheme can be applied in this case to perform decomposition, for example one-against-one class encoding (Masulli & Valentini, 2000). However, in the general case of heterogeneity with contextual features (Table 5) the groups of instances do not fall onto subsets of classes in the local regions. It is expected that local feature selection will result in simpler local models and better class separation.

Features possibly considered as contextual may not be present in a data set. This makes the search for the right splits very challenging. An adaptive search with instance weighting can be a possible solution for this problem. The difference in feature importance profiles after split can be used as an evaluation function for candidate decompositions.

A crucial distinction of the decomposition approach based on local feature relevance evaluation is that the evaluation function is based on a feature merit measure or a feature subset merit measure. An evaluation function applied to subsets of contextual or primary features should take into account interactions between features (Apte *et al*., 1998). This kind of evaluation function is used in so called "non-myopic" feature selection / ranking methods measuring the contribution of features to class discrimination that take into account interactions between features. The nature of feature interactions, that is their individual and mutual contribution to class discrimination, can be different. Thus, it is important to take into account the way a certain method considers these interactions. This is mostly related to the feature ranking methods that create a profile of feature importance in a local region.

Chapter 4 is dedicated to estimation of local feature relevance. It details the approach based on local feature relevance evaluation considering different feature selection and ranking methods along with feature weighting schemes. An evaluation function and a search strategy for this approach are proposed.

3.2.3 Decomposition based on local class separability estimation

Classification performance is directly influenced by class separability along with the other factors, such as dimensionality, training data size, and classifier type (Hsieh & Landgrebe, 1998). There are techniques attempting to improve class separability directly by changing data representation or using class separability in construction of local models. In particular, the approach proposed in this thesis is a combination of the above.

A natural and straightforward approach to construct a predictive model for heterogeneous data is to perform decomposition approximating homogeneous regions in data and modeling them separately (Apte *et al*., 1998). The domain knowledge in form of contextual features could guide such decomposition. For example, different diagnostic models may be built on different symptoms (features) for different genders or age groups, which can be considered as contextual features. However, the relevant domain knowledge for decomposition is often missing or not so obvious, especially because data not always purposefully collected for analysis. In this case, decomposition can be performed searching for local regions and evaluating complexity of classification locally (Ho & Basu, 2002). In particular, the BDP method proposed in Chapter 5 uses a criterion based on the class separability measure for evaluation of the candidate local regions. A heuristic search strategy needed to avoid the complete enumeration of the candidate local regions for evaluation is proposed as a *k*-Nearest Neighbor search for local neighborhoods in order to adjust weights.

A class separability measure can be derived from the complexity measure that characterizes local clustering properties of a data sample with respect to class labels (Ho, 2002; Pranckeviciene *et al*., 2006). This measure compares the dispersion within the classes to the separation between the classes.

Considering the two-class data case, let us denote $d_{l,same}$ the *intra-class distance*, which is calculated as an average distance between two nearest neighbors of the same class inside the group l. Let $d_{l,diff}$ be the *inter-class distance*, which is calculated as an average distance between two nearest neighbors of different classes inside the group l. The component distances calculated in one feature j can be denoted as $d^j_{l,same}$ and $d^j_{l,diff}$.

In (Ho & Basu, 2002) the ratio of the inter class and the intra class distances is used as a data complexity measure. Our goal is to maximize class separability in subproblems that is to maximize the ratio $\partial_l = \frac{d_{l,same}}{d_{l,diff}}$. A close to 1 ratio may be an indication of heavily interleaved classes.

Features that bring a considerable contribution to discrimination of classes can be determined using a measure based on the *overlap of the individual feature values* (OT) (Ho & Basu, 2002). For each feature $f_j, j = 1 \ldots N$, the minimum and the maximum value in class d is measured over M_l instances in the group l, $f^d_{j\,\min}, f^d_{j\,\max.}$ Then the measure of feature values overlap for classes 1 and 2 is

calculated as the length of the overlap region normalized by the range of values spanned by both classes (Formula 1).

$$OT_{1,2}^{j} = \frac{\min\left(f_{j\,\max}^{1}, f_{j\,\max}^{2}\right) - \max\left(f_{j\,\min}^{1}, f_{j\,\min}^{2}\right)}{\max\left(f_{j\,\max}^{1}, f_{j\,\max}^{2}\right) - \min\left(f_{j\,\min}^{1}, f_{j\,\min}^{2}\right)} \quad (1)$$

$OT_{1,2}^{j}$ is equal to 0 if the values of the particular features in two classes does not overlap. In order to evaluate the contribution produced by the instance i from the group l minimum and maximum values of feature j are found considering instance i in the group l and then withdrawing it from the group l, $\omega_i^j = \frac{OT_{1,2}^{j}(l|i)}{OT_{1,2}^{j}(l)}$. This measure is easily adapted to calculation of weights inside the relatively small groups and will be used in BDP.

Bidirectional Data Partitioning is based on the bottom-up search. It divides a data set into local regions to build local models simultaneously finding the feature subspaces such that groupings of instances from the same class tend to be denser, while different groupings tend to be far apart. The goal is to uncover the local structure in subspaces, where discrimination between classes will be simplified. The size of local regions is controlled to build the reliable local predictive models. In the overlapped subspaces, features that do not contribute to discrimination and grouping are assigned small weights by the choice of a weight function.

Theoretically, the solution is derived optimizing a criterion that encourages separability of instances from different classes and closer location of instances from the same class in feature subspaces. The weights assigned to features and instances are simultaneously adjusted. The process is repeated until all weights are stabilized. As a result, the local neighborhoods become increasingly enriched with instances belonging to the same region.

Final solution groups are modeled using relevant feature subspaces to produce the component classifiers of an ensemble. The rule of grouping instances with known class labels can be applied to new unclassified instances with a simple modification. Unclassified instances are assigned to one or another region based on the proximity of grouped instances, where proximity is determined only by distance disregarding class membership. Group memberships for the unclassified instances are used to determine the component classifier to be used for classification.

3.2.4 R-IPA for contextual heterogeneity

In Apte *et al*. (1998) the difference in class probability distributions in subproblems was considered as one of the heterogeneity criteria. Alternatively, the Importance Profile Angle (IPA) has been proposed to determine the degree to which the importance of features varies between two subproblems.

The profile of feature importance in subproblems is defined by the rank of normalized feature merits obtained by the ranking method (ReliefF, Symmetrical Uncertainty, or Information gain). The profiles of importance for subproblems *A* and *B* are denoted by the vectors of merits $\left(M_{A,1}, M_{A,2}, \ldots, M_{A,N}\right)$

and $(M_{B,1}, M_{B,2}, \ldots, M_{B,N})$ correspondingly, where N is the number of features and M denotes a measure of feature importance or feature merit.

IPA is the angle formed by the two vectors in the N-dimensional space. Formula 2 defines the normalized IPA taking values between 0 and 1 (Apte *et al*., 1998).

If the feature is assigned a zero merit, it does not improve discriminating ability of a model. The IPA can be calculated for both discrete and continuous features. The large value of the IPA indicates that class heterogeneity is present.

$$IPA = \frac{2}{\pi} \arccos\left(\frac{\sum_{i=1}^{N} M_{A,i} M_{B,i}}{\sqrt{\sum_{i=1}^{N} M_{A,i}^2} \sqrt{\sum_{i=1}^{N} M_{B,i}^2}} \right) \quad (2)$$

The search for the candidate subproblems can be performed using a tree-like algorithm (Apte *et al*., 1998). In order to find a candidate split, splitting is performed by the values of every feature, and the feature producing the best split determined by IPA is placed in the tree node. The best split corresponds to the maximum IPA value.

This procedure searches for the contextual features considering them independently. It creates a hierarchy of contextual features according to the difference in subproblems created by the splits. Assuming that there are p contextual features, $0 \leq p \leq P$, one has to set a stopping criterion for the splitting process. In Apte *et al*. (1998) a stopping criterion is defined by a certain threshold t_p placed for the IPA values. The experiments suggest $t_p = 0.4$ as a default value that can be used for many data sets.

If there are no contextual features in the data set, the threshold will never be exceeded and no splits will be performed. In this case, one of the alternative search strategies proposed in Section 4.3 can be used to find the candidate decompositions.

The tree-like search proposed in Apte *et al*. (1998) cannot find the candidate splits in the case of higher order dependency between contextual features. Thus, the random tree will be used as a part of this decomposition approach. A random tree-like search considers K randomly chosen features at each node and the splits are performed according to combination of their values. This method will further be referred to as R-IPA.

In order to obtain the reliable IPA estimates for the candidate splits and subsequently build a model for a local region, the number of instances in a subproblem should not be too small. This is a perennial problem for all tree-like algorithms. Therefore, another threshold t_n is needed to indicate a minimum number of instances in a subproblem n, $t_n \leq n \leq N$. If $n < t_n$, the candidate split is rejected. This problem implies a restriction for a number of interacting contextual features considered in a subset, as the number of candidate splits increases with the increased number of contextual features, and the number of instances in candidate subproblems tends to be small.

IPA value can be computed only for a pair of the candidate subproblems. For features having more than two discrete values, IPA can be computed for

each pair of the values considering splits between these two values and ignoring all instances, where this feature takes any other value. Then values can be merged recursively by grouping together the pair of values with the lowest IPA and regenerating the vector of feature merits for the just-grouped values versus the rest, until IPA values within each group are large comparing to the threshold t_p. As a result, more than two groups can be obtained. The smallest of the IPA values between the final groups should be used as an IPA for the feature being tested. (Apte *et al*., 1998)

It may happen that within a good split for one-class heterogeneity some features may have constant values. In decomposition, the feature having constant value for a subset of instances of the same class does not bring any new information and can be discarded.

Contextual dependence between contextual and primary features may be hierarchical, that is one set of contextual features controls relevance of another subset of features, which in their turn may serve as contextual features for some different subset of features. The original tree-like strategy from Apte *et al*., (1998) is preferable in this case.

The success of IPA estimates also depends on the feature merit measure used. Following the assumption that in local regions primary features are not necessary independent, the feature merit measure used in IPA calculation should take into account feature interactions. The observations from the experiments on the synthetic data sets described in Apte *et al*., (1998) confirm this.

3.3 Multi-model classification based on the decomposition scheme

Homogeneous regions are hard to elicit without prior domain knowledge, contextual features, or knowing that class combinations are worth exploring. That is general case called feature space heterogeneity in this thesis. For example, in (Golub *et al*., 1999), 4 subclasses of two leukemia cancer types were discovered, but an expensive additional study with production of additional features was needed to understand the meaning of those 4 subclasses.

Known methods capable to identify homogeneous regions, partially considered in Section 3.2, are designed for specific cases of heterogeneity, for example, heterogeneity influenced by the presence of contextual features. These points provide an argumentation for application of an ensemble framework in order to find a solution for the classification heterogeneity problem. Decomposition using ensemble is also well in concordance with the basic methods of ensemble generation: sampling, multiple feature subspace projection, and class encoding.

The main idea of applying combined ensemble generation methods for heterogeneity decomposition is the following:

- make multiple feature subspace projections on different instance sets and re-label a class variable in order to view classification subproblems in the relevant dimensions;
- apply to the training data to cover those subproblems by different learning models;
- apply the frames to the test data in order to identify the corresponding component classifier(s), or the combined classifier, for prediction.

Thus, the component classifiers can be generated combining sampling, multiple feature subspace projection and class encoding techniques. Searching through different subsets (samples) of instances and viewing them in different feature subspace projections in order to find homogeneous regions is an intractable, and sometimes an unfeasible task. Heuristics used in traditional ensemble techniques to narrow this search space may provide a solution for the problem.

3.3.1 Decomposition scheme for ensemble generation

First, let us consider a decomposition scheme for class heterogeneity. This is the simplest case when homogeneous regions include instances from particular groups of classes. From the above follows, that decomposition can be performed using class encoding.

Denote the *instance set decomposition matrix* which distributes D classes of the training set TR into two sets of classes A and B in the partition s as $DI = [z_{s,i}]$, where $s = 1 \dots S,\ S \le D,\ i = 1 \dots M,\ M$ is the number of instances in the training set TR. According to this matrix, S partitions will be created and thereafter a single component classifier h_s will be trained at each partition. The elements of the instance set decomposition matrix, each specifying which instance with what class label to use in sth classifier construction, are defined as shown in Formula 3.

$$z_{s,i} = \begin{cases} +1, \text{ if } y_i \in A \\ 0, \text{ if } y_i \notin (A \cup B) \\ -1, \text{ if } y_i \in B \end{cases} \tag{3}$$

The training set TR is sampled according to S partitions and in each sample the class variable is re-labeled according to the expression below.

$$y_i = \begin{cases} c_A, \text{if } z_{s,i} = +1 \\ c_B, \text{if } z_{s,i} = -1 \end{cases} \tag{4}$$

Denote the *feature set decomposition matrix* as $DF = [q_{s,j}]$, where $s = 1 \dots S$, $i = 1 \dots N$, N is the number of features in the feature set F used in the training set TR. This matrix specifies which P of N features, $P \le N$, from the initial feature set F to use in s^{th} classifier construction. In such a way, a subset $F' \subseteq F$ is used in each of S samples. The elements of the feature set decomposition matrix are defined as shown below.

$$q_{s,j} = \begin{cases} +1, \text{if } f_{s,i} \in F' \\ -1, \text{if } f_{s,i} \notin F' \end{cases} \quad (5)$$

The subsets F' of relevant features $f_{s,j}$ can be found by some feature selection method based on feature merit measures estimation. When individual feature merit measure calculations produce a rank of feature merits, some amount of features from the top of a rank can be used in construction of a component classifier. Different approaches to select a subset of features from rank are described in Section 4.2. Application of individual feature / feature subset merit measures for generating multiple classifiers has been preliminary investigated in Puuronen *et al.* (2001). The most appropriate individual feature / feature subset merit measures to use for local feature selection in subproblems are considered in the next chapter.

Class heterogeneity is a simple case of feature space heterogeneity (Section 3.1). It can be used for heterogeneous classification problems by analogy of the normal distribution that is successfully applied to data at many practical tasks. With application of class decomposition the task to find the instances corresponding to homogeneous regions comes to find subsets of classes. Decomposition for one-class and class heterogeneity is the most straightforward, but in order to complete this decomposition the subsets of relevant features have to be determined. Various individual feature / feature subset merit measures used in feature selection methods can be applied for this purpose. Several of them will be described in the next chapter.

Consider details of learning and application of an ensemble constructed using the decomposition matrixes above for one-class heterogeneity. In this case, subset A includes one class, and subset B includes the rest of classes. Such decomposition will be further referred to as *one-per-class decomposition*. The ensemble obtained as a result of such decomposition will be referred to as *one-against-all ensemble*. According to the instance and feature set decomposition matrixes, the component classifiers are constructed on locally relevant features and designed to distinguish a particular class from the others.

Each time a new component classifier has to be generated, the training set TR is partitioned assigning the representative instances to subsets A and B according to the instance set decomposition matrix and class encoding is performed re-labeling instances in a sample. Then some individual feature / feature subset merit measure is calculated at the re-labeled sample. According to measurements of individual feature / feature subset merit for each classifier $h_s,\ s = 1 \ldots S$, the feature set decomposition matrix is created and used to select a subset of features. In such a way, each component classifier is constructed on the re-labeled sample using selected subset of features. For one-per-class decomposition re-labeled sample includes all instances of the initial training set TR, while for the other class decomposition schemes it includes only instances from subsets A and B. So, for *pairwise class decomposition* both subsets A and B include only one class, thus only instances from those two classes are used to construct a component classifier. Ensemble created after pairwise class decomposition will be further referred to as *one-against-one ensemble*.

3.3.2 Integration of binary component classifiers

The component classifiers in one-against-all and one-against-one ensembles output two possible values, c_A and c_B. Predictions of the component classifiers can be considered as binary decisions that is to say, the classifiers are binary.

Integration of binary component classifiers in both one-against-all and one-against-one ensembles can be performed using probability calculations. This integration strategy is commonly used for ensembles based on class encoding, as for example, for pairwise coupling considered in Moreira and Mayoraz (1997).

According to this integration strategy for each new unclassified instance the component classifier output is the estimated membership probabilities that class variable y takes value c_i or c_j. Thus, a partial answer is provided regarding the class(es) that this component classifier is designed to distinguish. Considering the answers as votes a natural approach for integration is to choose the class that has received majority of votes. Assuming that the classifier discriminating between class c_i (as positive) and class c_j (as negative) computes an estimate of the probability $p_{i,j}$ as shown below.

$$p_{i,j} = P\left((x, y = c_i) | \left((x, y = c_i) \cup (x, y = c_j)\right)\right) \tag{6}$$

Then the ensemble prediction is determined according to the following expression.

$$\arg\max_{1 \le i \le D} \sum_{j \neq i} \eta(p_{i,j}) \tag{7}$$

In Formula 8 η is determined as follows.

$$\eta = \begin{cases} 0, \text{ if } p_{i,j} \le 0.5 \\ 1, \text{ otherwise.} \end{cases} \tag{8}$$

This integration strategy is a variation of *stacked generalization* (Wolpert, 1992). Ensemble prediction is obtained as an output of a stacked multiclass classifier. The mechanisms of accuracy improvement provided by stacking are based on information learned about biases each component classifier produces with respect to the initial classification problem (Tsymbal, 2002). The basic idea is to perform induction over the outputs of binary classifiers at higher level in order to correct their biases by means of a meta-classifier that combines predictions of the component classifiers.

This integration scheme is applied with one-against-all ensemble in the following way. For a new unclassified instance in reality belonging to class c_A the probability that it may belong to class c_j calculated by the outputs of S-1 component classifiers should be greater than 50%. But this probability should be smaller than probability that this instance belongs to class c_i, which is obtained from the output of classifier designed to distinguish class c_A from the other classes, that is $c_i = c_A$. In one-against-one ensemble for a new instance in reality belonging to class c_A D-1 component classifiers produce random

decisions; the other *S*-*D*-1 component classifiers should output high probabilities that the instance belongs to class c_A.

Ensembles based on one-per-class and pairwise class decompositions are able to provide accuracy improvement for classification tasks that are not heterogeneous (Masulli & Valentini, 2000). However, it was shown that error-correcting ensembles often outperform them due to redundancy in representation of the subproblems (Dietterich & Bakiri, 1995; Moreira & Mayoraz, 1998). Thus, for heterogeneous classification problems it is expected that extension of one-against-all and one-against-one ensembles with local feature selection by correcting codes will promote further accuracy increase as well as incorporation of boosting.

3.4 Chapter summary

In this chapter, decomposition approach and decomposition schemes are described. These schemes are intended for different types of heterogeneity and based on different evaluation functions and search strategies. Decomposition for construction of local models is performed assuming presence of data characteristics valuable to relate the classification problem to a particular type of heterogeneity.

Search for heterogeneity has to be performed in the feature space and in the set of instances simultaneously. However, assuming that some grouping of instances exists at the class level and estimating locally relevant features, candidate decompositions can be found and verified. In case of class heterogeneity, grouping of instances is related to class labeling. This is the simplest case of heterogeneity for which the class encoding decomposition scheme can be used.

Decomposition of heterogeneous classification problems performed within an ensemble framework is based on establishing domains of competence for the component classifiers in accordance with homogeneous regions found. In case of feature space heterogeneity, decomposition is performed at the level of subclasses or super classes.

4 DECOMPOSITION BASED ON LOCAL FEATURE RELEVANCE PROFILES

This chapter describes decomposition approaches focusing on evaluation criteria. Feature relevance profiles in subproblems obtained by means of feature merit measures are used for evaluation of the candidate subproblems. Class separability measures, widely applied to evaluate and score features in feature selection, are also considered here as suitable candidates to develop decomposition criteria.

Search strategies for the candidate subproblems are different in case of class heterogeneity, contextual heterogeneity, and feature space heterogeneity. Decomposition of class heterogeneity is based on the assumption that heterogeneity exists at the class level and the search is performed via class encoding, in this case, one-against-all and pairwise class combinations. The main idea of this approach is to find local regions out of different class combinations and establish features locally relevant to discriminate between subproblems. Local models are integrated as component classifiers of the ensemble by means of weighted voting.

Decomposition of contextual heterogeneity is based on a tree-like search procedure that performs splits according to values of contextual features. Evaluation function is based on the Importance Profile Angle (IPA), a measure derived from cosine similarity, applied to feature relevance profiles.

Feature merit measures producing ranks of feature scores, or feature relevance profiles, are applied to the candidate regions. The difference between feature importance profiles before and after decomposition obtained using an Importance Profile Angle (IPA) provides an additional evidence of

heterogeneity presence. Heterogeneity in presence of contextual features is also considered.

Section 4.1 discusses the effects of feature interactions and irrelevant features on class discrimination. It describes a suitable feature subset evaluation measure that takes into account feature correlations to be applied for local feature selection. Higher order dependency between features is presented as one of the characteristics attributed to heterogeneity. Contextual dependence between features is presented with respect to classification heterogeneity.

Section 4.2 is devoted to feature merit measures used in ranking methods applied either to select features or to obtain feature relevance profiles in candidate local regions.

4.1 Evaluation of individual features and feature subsets

Evaluation of features for the classification task is based on measures that express contribution of a feature or a subset of features to class discrimination. These measures include distance measures, information measures, dependence measures, or classifier error rate measures (Dash & Liu, 1997). In classification, the criterion for feature selection is related, but different from the criteria applied in clustering, because it incorporates information about classes. It is also somewhat different from the criteria used in dimension reduction, which are based on self-descriptiveness of the selected feature subset and measure how well the subset reproduces an intrinsic data structure (Aivazyan *et al.*, 1989).

Individual feature merit measures or subset merit measures are directly or indirectly related to the Bayes minimum error, therefore, many of them are based on class separability. Many class separability measures are estimates of Bayes minimum error. Feature selection techniques have a natural goal of minimizing the error using theoretical bounds or the error rate from a specific classifier.

Discovering interactions between features during evaluation of individual features or feature subsets is crucial, because these interactions reflect data structure. Features are considered interacting if dependence between the class variable and a feature is conditioned by the values of another feature. Interacting features may not be predictive individually, but predictive depending on the values of other features, which can be contextual features. Two or more features, whose ability to predict the class always depends on each other while their individual contribution to class discrimination is insignificant, exhibit higher order dependency. Higher order dependency may also exist between contextual features. It creates an additional obstacle to detect them.

A measured quantitative dependence between features gives an idea about the strengths of interaction using some numerical scale. Assigning interaction between features a score is desirable for many application tasks. In

particular, it can be used to find contextual features. Dependence between a feature and a class variable is a non-symmetrical relation. Dependence between redundant features is a symmetrical relation. Higher order dependence between features is a symmetrical relation as well. Dependence between a contextual and a primary feature is a non-symmetrical relation.

4.1.1 A correlation-based merit measure for a feature subset

Unknown relation between variables can be estimated using a modeled dependency in a standard form, for example, linear or nonlinear. In classification, these estimates are needed to decide whether one variable could be expressed by means of another. The term *correlation* can be applied here in its general sense referring to a degree of dependence or predictability of one variable to another. Statistical correlation coefficient is a measure of linear symmetrical relation between variables. For example, linear or nonlinear regression is a measure of a non-symmetrical relation between variables.

Considering influence of feature-feature relations on determining feature-class relations, the feature selection task is to retain individually predictive features, whose contribution to class discrimination does not depend on the other features and does not duplicate the information provided by the other individually predictive independent features of the same kind, and retain features that are not individually predictive but predictive together, in combination.

The similar idea formulated somewhat differently is used in the correlation-based feature subset selection (CFS) method proposed by Hall (1999). CFS is based on the idea that a good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other. Hall (1999) has compared performance of CFS to wrapper feature selection (Kohavi & John, 1998). It was shown that CFS is competitive to wrapper in many cases. Let us consider why the condition to discard correlated features was introduced in CFS.

In statistics, the correlation coefficient $r_{j,k}$ between two random variables (numerical features f_j and f_k) is found by dividing their sample covariance $S_{j,k}$ by the product of their sample standard deviations S_j and S_k, as shown in the expression below. (It is only defined if these standard deviations are finite.)

$$\mathrm{corr}_{j,k} = \frac{S_{j,k}}{S_j S_k} \tag{9}$$

where sample covariance $S_{j,k} = \frac{1}{M}\sum_{i=1}^{M}\left(x_i^j - \bar{x}^j\right)\left(x_i^k - \bar{x}^k\right)$ and sample standard deviations are $S_j = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}\left(x_i^j - \bar{x}^j\right)^2}$ and $S_k = \sqrt{\frac{1}{M-1}\sum_{i=1}^{M}\left(x_i^k - \bar{x}^k\right)^2}$.

The correlation takes values 1 or -1 in the case of an increasing or decreasing linear relationship accordingly. Values in between express the degree of linear dependence between the variables. If the variables are

independent the correlation is 0, but the converse is not true because the correlation coefficient detects only linear dependencies between two variables.

Pearson's product moment correlation coefficient $r_{j,k}$ (Kenney & Keeping, 1962) is also known as a sample correlation coefficient and defined as the sum of the products of the standard scores z_j and z_k of f_j and f_k divided by the degree of freedom. It is a measure of the linear association between two variables which have been measured on interval or ratio scales (Formula 10).

$$r_{j,k} = \frac{1}{(M-1)} \sum_{j=1}^{M} z_j z_k = \frac{\mathrm{corr}_{j,k}}{(M-1)} \tag{10}$$

where $z_j = \frac{x^j - \bar{x}^j}{s_j}$ and $z_k = \frac{x^k - \bar{x}^k}{s_k}$. However, Pearson's correlation coefficient can be misleadingly small when there is a nonlinear relationship between two variables.

The merit of a feature subset in CFS corr_F is calculated according to Formula 11, where F is a feature subset containing P features, $P < N$, y is a class variable, $\bar{r}_{j,y}$ is the average feature-class correlation $(F_j \in F)$, and $\bar{r}_{j,k}$ is the average feature-feature correlation.

$$\mathrm{corr}_F = \frac{P\left|\bar{r}_{j,y}\right|}{\sqrt{P + P(P+1)\bar{r}_{j,k}}} \tag{11}$$

This formula was initially developed by researches in behavioral sciences and adopted to feature selection purposes in data mining by Hall (1999). The subset merit measure in Formula 19 is derived from the Pearson's correlation coefficient, where all variables have been standardized (Hall, 1999). Standardized quantity is invariant over changes in the units of measurement. It turns out that the numerator in Formula 19 gives an indication of how predictive for the class the subset of features is, and the denominator expresses how much redundancy there exists. The formula shows that relation between a feature subset and a class variable is a function of the number of features in a subset and the magnitude of feature-feature correlations together with the magnitude of feature-class correlations (Hall, 1999).

The CFS merit measure implies that (1) the higher feature-class correlations in the subset, the higher subset score is, (2) the lower feature-feature correlations in the subset, the higher subset score is, and (3) the more highly correlated with class yet uncorrelated to each other features are included into the subset, the higher subset score is.

Features whose individual ability to predict a class variable always depends on the other features will be discarded by the CFS merit measure (Hall, 1999). This holds true for both redundant features, which are individually predictive, and interactive features with higher order dependencies, which are not individually predictive, or appear to be partially predictive. Thus, CFS can be used with a measure of linear or nonlinear dependence as a base correlation measure, but it cannot take into account higher order feature dependencies (Hall, 1999).

The CFS measure (Hall, 1999) is designed to take into account a certain type of feature dependencies defined by the basic correlation measure used in it. However, with some modifications proposed in Hall (1999), it and can be applied to estimate mutual relations between features taking into account higher order feature dependencies. These modifications and an alternative method are considered in the next subsection.

Statistical correlation coefficients provide an estimate for the strength of linear dependency between features or a feature and the class variable. In order to obtain a general measure for different dependencies in the data (also nonlinear) a correlation ratio or mutual information based measures is preferable. Such measures can be effective in finding dependencies of different kind.

A measure of predictive ability of a feature regarding a class variable is usually non-symmetrical. A symmetrical variant is needed to estimate mutual predictive abilities of features. Correlation between two features, or a feature and a class variable, can be evaluated using the Symmetrical Uncertainty (SU) measure (Hall, 1999), a variant of the Information gain measure.

By Hall (1999), the Information gain measure (IG) is biased in favor of features with more values. SU is a normalized version of measure proposed in Hall (2000) to estimate mutual predictability of two features (Formula 12). SU can be used to calculate correlation between categorical (nominal) variables as well as between numeric variables.

$$SU_H = 2\frac{IG}{H(y) + H(f)} \tag{12}$$

Another symmetrical measure proposed in Hall (1999) based on the Minimum Description Length principle (MDL) (Rissanen, 1978) is a modification of its non-symmetrical version developed by Kononenko (1995).

The measures applied to individual features are often called in the literature "myopic" if they do not take into account (indirectly) feature interactions during individual feature evaluation. IG, SU, the MDL-based measure, OT (Formula 1 in subsection 3.2.3) and Fisher's discriminant ratio are examples of "myopic" measures. There are measures called "non-myopic" that consider an impact of feature interactions indirectly while evaluating individual contribution of a feature to class discrimination. The examples of such measures is ReliefF merit measure considered in Subsection 4.3.2.

Non-symmetrical "non-myopic' measures, just as "myopic" measures, can be transformed to its symmetrical variants. For example, symmetrical variants of the measures investigated by Kononenko (1995) can be applied to calculate mutual relations between features.

The statistical correlation coefficients are applicable to numeric features only. However, the data used for classification tasks is usually a mixture of categorical (nominal) and numeric (discrete and continuous) variables. In order to calculate correlation for discrete variables binarization of discrete variables may be performed, for example, as proposed in Hall (2000).

Let us consider f_j being a discrete feature having T values $v_1^j, \ldots, v_t^j, \ldots, v_T^j$. One may form T binary features b_t, $t = 1 \ldots T$ according to Formula 13.

$$b_t = \begin{cases} 1, \text{if } f_j = v_t^j \\ 0, \text{otherwise} \end{cases} \tag{13}$$

The expression below is used to calculate correlation between the discrete feature f_j transformed to T binary features b_t and a continuous feature f_k.

$$r_{j,k} = \sum_{t=1}^{T} P\left(f_j = v_t^j\right) r_{t,k} \tag{14}$$

If f_j is a discrete feature having T values $v_1^j, \ldots, v_t^j, \ldots, v_T^j$ and f_k is a discrete feature having L values $u_1^j, \ldots, u_l^j, \ldots, u_L^j$, then the binary features f_j^t, $t = 1 \ldots T$ and f_j^l, $l = 1 \ldots L$ are formed as described above. Correlation between these two discrete features can be calculated using the following expression (Hall, 2000).

$$r_{j,k} = \sum_{t=1}^{T} \sum_{l=1}^{L} P\left(f_j = v_t^j, f_k = u_l^j\right) r_{t,l} \tag{15}$$

The above two expressions are robust to missing values (Hall, 2000).

4.1.2 Higher order dependency between features

In some cases, the ability of a particular feature to predict the class variable always depends on the other features as measured on a particular sample or the entire training set. This is called a *higher order dependency* between features. Contrary to redundant features, two features having higher order dependency are not predictive of the class individually.

The synthetic two-spirals classification problem (Lang & Witbrock, 1988) shown in Figure 7 is an example of interacting features with a nonlinear dependency. Features f_1 and f_2 are predictive only if considered together. The data has 3 irrelevant features f_3,f_4, and f_5, uniformly distributed in the interval [0..12]. Figure 8 shows 3 dimensions corresponding to features f_1, f_4 and f_5. The presence of interacting features negatively affects accuracy of many learning algorithms. Irrelevant features destroy the data structure, as shown in Figure 8.

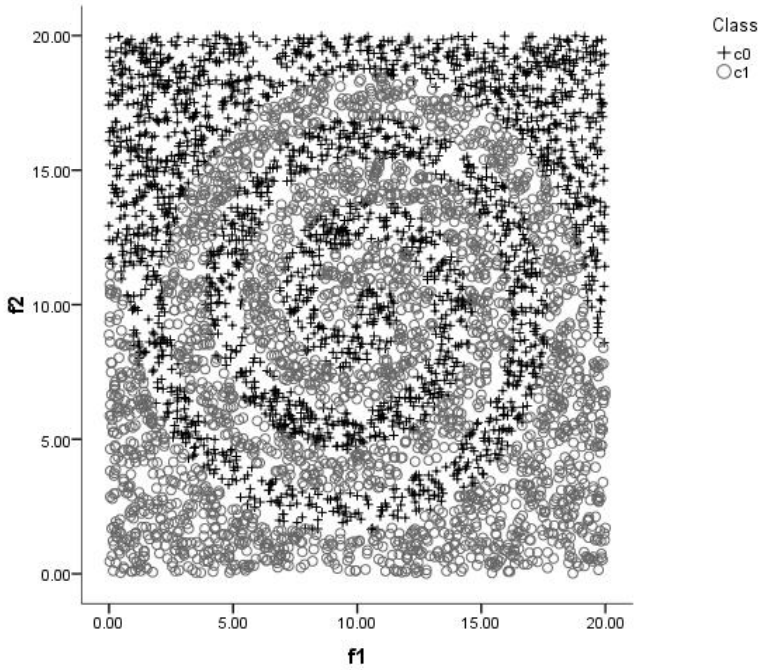

FIGURE 7 Two spirals data set in 2-dimensional feature space (Lindenbaum *et al.*, 1999). Features f_1 and f_2 are not correlated in terms of statistical correlation tests (Pearsons and Spearman's correlation coefficients are 0.003, Kendall's correlation coefficient is 0.002). However, these features are considered as interacting regarding their contribution to class discrimination.

Detecting higher order dependencies in general is difficult because of the various problems with data, in particular, imbalanced class representation that increases risk of overfitting and mixed feature types that require discretization and may result in biased estimates.

Evaluation of all possible combinations of feature subsets for dependency is an intractable task. In Sahami (1996), it is shown that even when each feature is constrained to be dependent on at most two other features the search for interacting features is an NP-hard problem.

In Hall (1999), a limited pairwise approach to detect feature interactions is presented. This approach is straightforward and computationally feasible. A similar approach to detect higher order feature dependencies during feature subset selection is proposed below.

There are two features f_i having T_i values $v_{1,i}, \dots, v_{t,i}, \dots, v_{T,i}$ and f_j having T_j values $v_{1,j}, \dots, v_{t,j}, \dots, v_{T,j}$. Joining these features a derived feature $z_{i,j}$ having $T_i T_j$ values is obtained. An algorithm that considers all possible pairwise combinations of features in this manner is quadratic to the original number of features (Hall, 1999). Once a derived feature is created, its predictive ability regarding the class variable can be calculated using a feature merit measure.

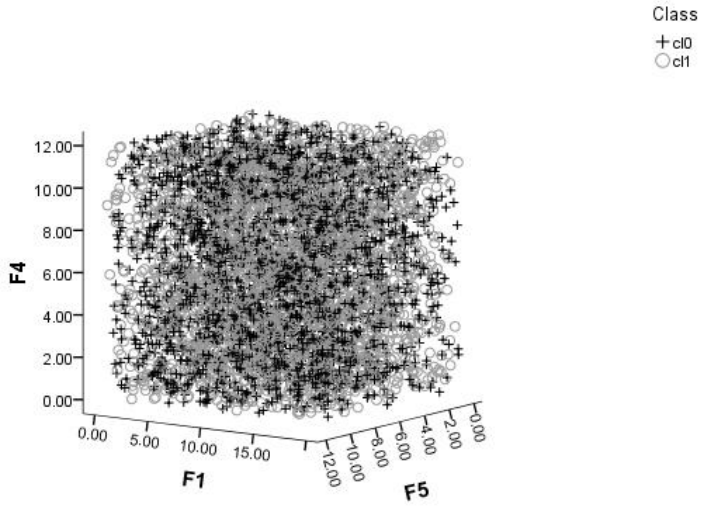

FIGURE 8 Two spirals data set in 3-dimensional feature space (Lindenbaum *et al.*, 1999). Features f_4 and f_5 are uniformly distributed irrelevant features. In this projection the spiral data structure is destroyed.

Possible strategies include evaluation of the non-symmetrical relation between derived features and the class variable, non-symmetrical relation between individual features and the class variable and symmetrical relation for pairs of features. These symmetrical and non-symmetrical relations can be calculated using a particular feature merit measure and its symmetrical version. In this thesis, the experiments are performed using ReliefF an SU measures. The basic steps are:

- evaluation of individual contribution to class discrimination for each feature to determine strongly predictive relevant features;
- pairwise evaluation for strongly predictive of each other redundant features;
- pairwise evaluation for higher order dependency between features joining their values and comparing the predictive ability of a derived feature to individual predictive abilities of two features;
- confronting results of the previous steps to select the individual predictive features, predictive features of higher order dependency and to discard the redundant features.

The threshold, t_m (default practical value is $t_m = 0.6$) specifies which features are considered as relevant according to the normalized feature merit M_f [0..1]. If $M_f > t_m$, the feature is relevant. Then pairs of relevant features are evaluated for redundancy. Features assigned $M_f < t_m$ are considered in pairs and the derived features are evaluated same as the individual features.

This method can be applied to the subset of instances representing a subproblem found after decomposition. The decomposition is performed using IPA. If IPA cannot find contextual features for splits, an alternative weighting scheme can be used to perform decomposition.

4.1.3 Contextual dependence between features

The *contextual dependence* between features can be considered as a particular case of higher order dependency. Domingos (1997) presents the concept of local feature relevance considering contextual dependence. By his definition, *some features may be highly relevant in certain regions of the instance set being irrelevant everywhere else by their sensitivity to a context, that is to the values of the other features*.

In Turney (1996, 1993), the definition of *primary*, *contextual* and *irrelevant* features is proposed. By this definition, primary features are useful for classification when considered in isolation, without regard for the other features. Contextual features are not useful in isolation, but can be useful when combined with other features. Irrelevant features are not useful, either when considered alone or when combined with other features. In this definition, contextual features are defined as having higher order dependency between features in general.

Harries and Horn (1996) define so-called *environmental* features, which reflect hidden context, for example, time or spatial location. The SPLICE method described in Harries and Horn (1996) does not select or extract the contextual features from heterogeneous data, assuming that context may be contiguous over some features (environmental).

In this thesis, the notion of *contextual dependence* is used to address the situation when relevance of a particular feature, called *contextual*, depends on the values of another feature, called *primary*. If contextual features are known in advance or can be identified, they can be used to perform decomposition by their values and find locally relevant features in subproblems.

In Domingos (1997), a *feature difference* measure is proposed to evaluate the contextual dependency. The Relevance-in-Context method described in Domingos (1997) is distance-based and instance-specific that makes it computationally expensive.

Harries and Horn (1996) have proposed a meta-level algorithm that uses a learning algorithm capable to perform context-sensitive feature selection, similar to the Relevance-in-Context method, or decision tree. Commonly, these methods perform selection of features at each node, rule, or clause in the context of locally relevant prior selections. Partitions are made over a contextual feature. Then contextual clustering is performed over the intervals according to apparent similarity of context, and local context-specific concepts are learned.

The presence of contextual features can be foreseen by considering individual and joint distribution of feature values. Similar feature evaluation is performed in a hierarchical density-based subspace clustering technique (Parsons *et al*., 2004). A random tree-like procedure using IPA suggested to

determine contextual features and perform splits (Apte *et al*., 1998) is used as a prototype for R-IPA search for decomposition of contextual heterogeneity described in Subsection 3.2.4.

4.2 Estimation of dissimilarity between subproblems

This section presents feature merit measures that can be used in evaluation of candidate local regions as well as in the subsequent feature selection. Those are feature ranking methods that assign each feature a numerical score. Dissimilarity between subproblems is measured by means of Importance Profile Angle (IPA) that uses numerical feature scores in a vector form. Alternatively, those vectors can be made of feature weights, as in the BDP technique, proposed in Chapter 5.

Some measures evaluating individual feature's contribution also can foresee the effects caused by interaction with other features. Such measures are often called "non-myopic" measures. The example is ReliefF (Kononenko, 1994) and contextual merit measure (Hong, 1997; Skrypnyk, 2005). In this section, two individual feature merit measures are considered. Local feature selection in the subproblems can be performed after decomposition using those measures and the correlation-based measure, presented in Subsection 4.1.1.

In feature selection, and individual feature or a feature subset are evaluated. Estimating a quality of a subset is more advantageous, because even measures that indirectly take into account interactions between features do not perceive all kinds of higher order relations that may exist between features. However, for high-dimensional data evaluation of subsets is computationally expensive. Exhaustive search through all possible feature subsets with a fixed number of features becomes intractable. Therefore, a heuristic search procedure is usually applied (Dash & Liu, 1997).

There are two approaches to measure merit of an individual feature. The first approach, often called "myopic", is based on the assumption about independence of features; therefore, the merit of a feature is estimated ignoring the other features. The second approach, called "non-myopic", considers feature interactions; therefore, the merit of a feature is estimated taking into account the values of the other features. A "myopic" approach is computationally more efficient than a "non-myopic" one, but the latter has a potential in discovering relations between features and higher order relations (Kononenko & Hong, 1997; Hall, 1999).

Interactions between features can be measured similar to measuring feature's contribution in discrimination between classes, but using symmetrical variants of "myopic" measures. Non-symmetrical and symmetrical variants of a "myopic" mutual information based measure, Information gain, are considered in Subsection 4.2.1. A "non-myopic" feature merit measure, ReliefF, is described in Subsections 4.2.2.

More advanced feature subset selection methods take into account a certain kind of feature interactions. The example is the correlation-based feature subset selection (CFS) method (Hall, 1999) described in Subsection 4.1.1. The correlation-based merit measure can be considered as a shell for using symmetrical and non-symmetrical "myopic" measures.

To some extent, all feature merit measures are biased toward the number of discrete values, classes, number of training instances, and so on. It is important to take into account those biases to understand the effect of feature selection as a part of an ensemble technique. Different ways to measure biases of various feature merit measures have been explored in Hong *et al.* (1996), White & Liu (1994), Kononenko (1995), and Hall (1999). Feature merit measures considered in this work are reported to have reasonable biases.

4.2.1 The mutual information based measures

In machine learning, the mutual information (information gain, or cross entropy, or in probability theory, Kullback-Leibler divergence) measure is a widely used information theoretic descriptor for stochastic dependency of discrete random variables (Kullback, 1968; Cover & Thomas, 1991; Soofi, 2000). Mutual information estimates the general dependence of random variables without making any assumptions about the nature of their underlying relationships. It is used to select features for classification problems on the basis of low values of mutual information with the class (Zaffalon & Hutter, 2002; Duda *et al*., 2001).

The mutual information based measures for estimation of individual feature merit considered here originate to information theoretic impurity measures used in decision tree induction, Information gain (Quinlan, 1986), and Gini (Breiman *et al*., 1984). Information theoretic measurements, like entropy, are able to express an ability of a feature to distinguish among several classes.

These measures are "myopic" assuming independence of features. Many data mining algorithms are based on this assumption. These methods are applicable to many classification problems, where interactions between features have no or only marginal effect (Kononenko *et al*., 1997). However, this assumption is often violated for heterogeneous data sets, which are in focus of this research.

Using the mutual information criterion feature merit is measured by the difference between an impurity of the classes and the resulting impurity of the classes under assumption that the feature value is known. The most frequently used in decision tree induction impurity measures are entropy used in C4.5 decision tree and Gini index used in CART decision tree algorithms.

In general, impurity can be determined as a function of a set of probabilities, which are summed up to 1. Below impurity measures will be defined similar to Apte *et al*. (1998) using the notions provided in Section 2.1.

Let us denote by $I(P_1, \ldots, P_K)$ the impurity of a set of probabilities $P_1, \ldots, P_K$, $P_1 + \cdots + P_k + \cdots + P_K = 1$. An impurity measure should satisfy

$I(P_1, \ldots, P_K) = 0$ whenever $P_k = 1$ for some k, and should be maximized when $P_k = 1/K$ for all k.

The entropy measure of impurity is then defined according to the following expression.

$$H(P_1, \ldots, P_K) = -\sum_k P_k \log_2 P_k \tag{16}$$

A probabilistic model for a categorical (nominal) class variable y, or a categorical feature f, can be formed by estimating the individual probabilities of the values it takes, $y \in (c_1, \ldots, c_d, \ldots, c_D)$ or $f \in (v_1, \ldots, v_t, \ldots, v_T)$ respectively, from the training data TR, where D is the number of classes, T is the number of different values of a feature. Consider y taking $c_1, \ldots, c_d, \ldots, c_D$ values with frequencies $\eta_1, \ldots, \eta_d, \ldots, \eta_D$ estimated from TR, and f taking $v_1, \ldots, v_t, \ldots, v_T$ values with frequencies $\mu_1, \ldots, \mu_t, \ldots, \mu_T$ estimated from TR. The conjunction of the class value c_d and the feature value v_t occurs with frequency $\omega_{d,t}$.

The impurity of the class variable can be defined as shown in Formula 17, where M is a number of instances in TR.

$$I(f) = I(\eta_1/M\,, \ldots, \eta_d/M\,, \ldots, \eta_D/M) \tag{17}$$

The impurity of the feature is then defined by Formula 18.

$$I(f) = I(\mu_1/M\,, \ldots, \mu_t/M\,, \ldots, \mu_T/M) \tag{18}$$

For instances from TR corresponding to a particular value v_t of a considered feature $f_j = f$, the impurity is denoted according to Formula 19.

$$I(y|f = v_t) = I\big(\omega_{1,t}/\mu_t, \ldots, \omega_{d,t}/\mu_t\,, \ldots, \omega_{D,t}/\mu_t\big) \tag{19}$$

When feature value v_t occurs with probability $P(v_t)$ the average impurity of a class variable y, given the feature f can be written as follows.

$$I(y|f) = \sum_{t=1}^{T} P(v_t)\, I\big(\omega_{1,t}/\mu_t, \ldots, \omega_{d,t}/\mu_t\,, \ldots, \omega_{D,t}/\mu_t\big) \tag{20}$$

For the training set of Minstances the proportion of occurrences of a feature value v_t is μ_t/M. Using this value as $P(v_t)$ Formula 17 can be expressed as shown in Formula 21.

$$I(y|f) = \sum_{t=1}^{T} \frac{\mu_t}{M} I\big(\omega_{1,t}/\mu_t, \ldots, \omega_{d,t}/\mu_t\,, \ldots, \omega_{D,t}/\mu_t\big) \tag{21}$$

This is impurity remaining in a class variable after the information present in the particular feature variable has been used.

If the observed values of y in TR are partitioned according to the values of a particular feature f, the entropy of y with respect to the partitions induced by f, $H(y|f)$ is less than the entropy of y prior to partitioning, $H(y)$.

The average impurity of a class variable y, after observing the feature f using the entropy can be written as follows.

$$H(y|f) = -\sum_{t=1}^{T}\frac{\mu_t}{M}\sum_{d=1}^{D}\frac{\omega_{d,t}}{\mu_t}\log_2\frac{\omega_{d,t}}{\mu_t} = \frac{1}{M}\left[\sum_{t=1}^{T}\mu_t\log_2\mu_t - \sum_{t=1}^{T}\sum_{d=1}^{D}\omega_{d,t}\log_2\omega_{d,t}\right] \quad (22)$$

The best feature is the one that achieves the lowest value of the entropy $H(y|f)$.

The amount by which the entropy of y decreases reflects additional information about y provided by f. This can be used as a measure to estimate merit of a feature f.

The Information gain (IG) measure is defined according to the following expression.

$$IG(y|f) = H(y) - H(y|f) = H(f) - H(f|y) = H(y) + H(f) - H(y,f) \quad (23)$$

According to this measure, a feature y is more correlated to feature f_1 than to feature f_2, if $IG(y|f_1) > IG(y|f_2)$.

Formula 23 demonstrates that Information gain is symmetrical measure, that is the amount of information gained about y after observing f is equal to the amount of information gained about f after observing y. Hence, these measures can be used to measure interactions (correlation) between two categorical features.

Information gain is biased in favor of features with more values. Thus, the values have to be normalized to ensure they are compatible and have the same affect. Symmetrical Uncertainty (SU) considered in Section 4.1 is a symmetrical version of Information gain that compensates for Information gain's bias. Symmetrical Uncertainty also normalizes Information gain's values to the range [0,1] with the value 1 indicating that knowledge of the value of either one completely predicts the value of the other and the value 0 indicating that two features are independent. (Yu & Liu, 2003).

In the C4.5 decision tree learning algorithm Gain ratio is used instead of Information gain, because the latter tends to favor features with large number of values. Gain ratio (GR) is defined according to Formula 24.

$$GR(y|f) = \frac{IG(y|f)}{H(f)} = \frac{H(y) + H(f) - H(y,f)}{H(f)} \quad (24)$$

4.2.2 The ReliefF measure

The ReliefF measure is derived from the Relief algorithm developed by Kira and Rendell (1992a and 1992b) and its extension ReliefF (Kononenko, 1994) for estimating merits of features with strong dependencies among them. In contrast to "myopic" feature merit measures, such as Information gain and similar estimates like Gini index or Mántaras's distance measure (Lopez de Mántaras, 1991), Relief takes into account dependency between features estimating feature merits. Features are evaluated according to how well their values distinguish between instances that are near each other in terms of Relief's distance function. In Relief for a given instance *two* its nearest neighbors are taken: one from the

same class (called *nearest hit*) and another from a different class (called *nearest miss*). The original Relief's estimate of feature merit is limited to *two* nearest neighbors and to the *two*-class problem. For a given instance (x_r, y_r), $r = 1 \ldots M$, where M is the size of TR, the merit of feature f_j, $RF(f_j)$, is estimated as shown below.

$$RF(f_j) = P(x_{k,j} \neq x_{r,j} | y_k \neq y_r) - P(x_{k,j} \neq x_{r,j} | y_k = y_r) \tag{25}$$

Instances (x_k, y_k), $k = 1 \ldots K$, are the nearest neighbors of (x_r, y_r) in terms of Relief's distance function. In the original Relief method Kequals to 2. The distance function in Relief is determined as a difference between values of particular feature for instances of different and same classes. The distance $d_{r,i}^{f_j}$ between the values of a discrete feature f_j for the given instance (x_r, y_r) and an instance (x_i, y_i) from different or the same class, that is $y_r = y_i$ or $y_r \neq y_i$ is defined as shown in Formula 26.

$$d_{r,i}^{f_j} = \begin{cases} 0, \text{ if } x_{r,j} = x_{i,j} \\ 1, \text{ otherwise} \end{cases} \tag{26}$$

The distance $d_{r,i}^{f_j}$ between the values of a continuous feature is defined as shown in Formula 27.

$$d_{r,i}^{f_j} = \frac{|x_{r,j} - x_{i,j}|}{f_{j\,\max} - f_{j\,\min}} \tag{27}$$

This $d_{r,i}^{f_j}$ is used for calculating the distance between instances to find the nearest neighbors. The total distance $D_{r,i}$ defined below is simply the sum of distances over all features, so-called Manhattan distance, where N is a number of features.

$$D_{r,i} = \sum_{j=1}^{N} d_{r,i}^{f_j} \tag{28}$$

Relief updates a merit over all features for a particular instance from the training set depending on their values for the given instance, its nearest hit, and its nearest miss. Let us denote the nearest neighbor of the instance (x_r, y_r), $r = 1 \ldots n, n \leq M$, from the same class (nearest hit) as (x_h, y_h), $h = 1 \ldots K$, and from the different class (nearest miss) as (x_m, y_m), $m = 1 \ldots K$. Then for every instance randomly selected from the training set (or a sample) the Relief merit measure is recursively calculated according to the following expression. Initially $RF(f_j)$ is set to 0.0, and then updated over all features, $j = 1 \ldots N$. For any new instance selected from the training set Relief merit measure is updated. Normalization by n guarantees that all merits are in the interval [-1, 1].

$$RF(f_j) = RF(f_{j-1}) - d_{r,h}^{f_{j-1}}/n + d_{r,m}^{f_{j-1}}/n \tag{29}$$

The rationale is that a good feature should differentiate between instances from different classes and should have the same value for instances from the

same class. If the instances (x_r, y_r) and (x_h, y_h) have different values of the feature f_{j-1} then feature f_{j-1} separates the two instances from the same class, which is not desirable, and therefore, the merit is decreased. On the other hand, if the instances (x_r, y_r) and (x_m, y_m) have different values of the feature f_{j-1} then feature f_{j-1} separates the two instances from different classes, which is desirable, then the merit is increased. The process is repeated n times, where n is user defined parameter, $n \leq M$. In order to get more precise estimate n can be set to its upper bound M, the number of instances in the training set TR.

In Kononenko (1994) an extension of Relief for multi-class problems, ReliefF, is proposed. This extension takes into account k nearest neighbors, and can deal with missing feature values and noisy data as well (Kononenko, 1994; Robnik-Šikonja & Kononenko, 2003). ReliefF searches for k nearest hits and k nearest misses from each of the different classes. The contribution of all the hits and all the misses is averaged. The contribution for each class of the misses is weighted with the prior probability of that class $P(y_i = c_d)$, $d = 1 \ldots D$, (estimated from the training set). The ReliefF merit measure (RFF) is derived from Formula 30 as shown below (Robnik-Šikonja & Kononenko, 2003).

$$\mathrm{RFF}(f_j) = \mathrm{RFF}(f_{j-1}) - \frac{1}{n \cdot k}\sum_{i=1}^{k} d_{r,h_i}^{f_{j-1}} + \frac{1}{n \cdot k}\sum_{y_m \neq y_r}\left[\frac{P(y = y_m)}{1 - P(y = y_r)}\sum_{i=1}^{k} d_{r,m_i}^{f_{j-1}}\right] \tag{30}$$

The selection of k hits and misses is the basic difference to Relief, which ensures greater robustness of ReliefF concerning noise. The k is the user-defined parameter that controls the locality of the estimates. For most purposes it can be safely set to 10 (Kononenko, 1994; Robnik-Šikonja & Kononenko, 2003).

In order to deal with missing feature values the distance function can be changed as shown in the following two expressions. Formula 31 is applied for the case when one instance (x_r, y_r) has unknown feature values. Formula 32 is applied for the case when both instances (x_r, y_r) and (x_k, y_k) have unknown values. Then the probability that two given instances have different values for the given feature conditioned over class is calculated according to the following expressions. Consider f_j having T values $v_{1,j}, \ldots, v_{t,j}, \ldots, v_{T,j}$.

$$d_{r,k}^{f_j} = 1 - P(x_{k,j} | y = y_r) \tag{31}$$

$$d_{r,k}^{f_j} = 1 - \sum_{t=1}^{T}\left(P(v_{t,j} | y = y_r) P(v_{t,j} | y = y_k)\right) \tag{32}$$

The conditional probabilities are approximated with the corresponding frequencies from the training set.

The context sensitivity in the RFF measure is provided by the "nearest instance" condition. The key idea is to estimate feature merits according to how well their values distinguish between the instances that are near each other.

Kira and Rendell (1992) provide experimental evidence that Relief is effective at identifying relevant features even when they interact, for example, in parity (XOR) problems. However, according to Kira and Rendell (1992), if

most of the given features were relevant to the concept, Relief would select most of the given features even though only a small number of them is necessary for concept description. In other words, Relief does not discard redundant features.

In Hall (1999), the symmetrical version of ReliefF has been developed and applied for correlation-based feature subset selection. To use ReliefF symmetrically for two features, the measure is calculated twice, so that each feature is treated in turn as a dependent variable, and the results are averaged. Formula 33 shows symmetrical version of ReliefF (SRFF).

$$SRFF(f_1|f_2) = \frac{RFF(f_1) + RFF(f_2)}{2} \tag{33}$$

4.2.3 Biases of feature merit measures

Information gain, ReliefF and CM suffer from the inherent bias that favors features having more values (Hall, 1999; Hong *et al*., 1996; Yu & Liu, 2003). White and Liu (1994) have demonstrated the prevalence of this bias and discussed the negative effects that it has on the predictive models that were constructed using biased measures. In Hong *et al*. (1996) the term *variety effect* were used to describe this phenomenon, especially in connection with numeric features that have mostly unique values for each instance in the training set.

When a feature takes many distinct values, it certainly has more power to model the target variable (class). At the extreme case, a feature whose values are distinct for training instances is sufficient by itself to model the target variable. In real data sets, it happens often in the form of ID features, account numbers or names. Such features should be excluded at the pre-processing step.

In Hall (1999) a systematic study of the bias caused by the number of feature values was performed along with the study of effects caused by small training set size for Information gain, ReliefF and their symmetrical variants. It was concluded that the effect tends to increase for small training sample sizes. For irrelevant features with many values this is especially undesirable, because such feature will appear more useful than a relevant feature with fewer values.

In practical terms, feature selection using these measures should prefer features with fewer values to those with more values. Since probability estimation is likely to be more reliable for features with fewer values, especially if the training set size is limited, there is less risk of overfitting the training data. (Hall, 1999)

In Hong *et al*. (1996) a scheme based on randomization was proposed in order to neutralize this bias by normalizing feature merits. The basic idea is to consider what would be the merit for a random feature with the same distribution of values as a given feature. Then, if the merit of the original feature is close or less than the average merit of random features with the same distribution, the feature should not be assigned a high merit value. A random feature with the same distribution is obtained by random permutation of the original feature's among the training instances. The original feature's merit is

then normalized dividing it by the expected merit of the random feature. A feature in question, whose merit is lower than a merit of random feature, is a candidate to be removed. However, "non-myopic" feature merit measures reflect a discriminative power of the feature in the presence of other features. It means, if the normalized value is less than one, a random feature would contribute more in the presence of other features. Hence, additional information about "how much better than random" should be introduced. (Hong *et al.*, 1996)

There could be alternative normalization schemes. For example, in Yu and Liu (2003) it was indicated that Symmetrical Uncertainty somewhat compensates bias toward features with more values and normalizes merit values to the range [0,1], where 1 indicates that feature has maximum discriminative power for the target variable. In Hall (1999), it was shown that symmetrical variants of Information gain and modified ReliefF ("myopic" version with removed context sensitivity provided by nearest instances) measures also reveal bias toward features having many values and to the small training set size. The estimates of both Symmetrical Uncertainty and modified Symmetrical ReliefF show a tendency to increase exponentially with fewer training instances. The effect is more marked for features with more values.

4.3 Class separability based and other complexity measures

Class separability in a feature subspace characterizes class discrimination using a variety of measures not related directly to a Bayes error estimate. These measures reflect statistical, geometrical and topological, or information-theoretic properties of data. Majority of class separability measures are primarily used in unsupervised learning, as they reflect a criterion for clustering.

4.3.1 Class separability and Bayes minimum error

Assuming that distributions of features as random variables are known, Bayesian classifier is a theoretically best classifier. It minimizes the probability of misclassification and therefore has the smallest possible error (Bayes minimum error, ε). It can be viewed as a cardinal class separability measure as by definition it is a minimal classification rate that can be obtained for a certain data (Fukinaga, 1990; Pierson, 1998). In most cases, calculation of ε is not feasible, because it relies on probability density functions in each class, prior class probabilities, and requires numerical integration. In case the only information regarding underlying distributions is a finite data sample, estimation of the probability density functions is made and the upper bound for ε is found. Difficulties in estimating ε has led to development of different class separability measures based on mathematical bounds on ε, information-theoretic concept of class separability, nonparametric proximity measures, and heuristic concept of class separability (Pierson, 1998).

Class separability in a feature subspace characterizes class discrimination using a variety of measures. Some of them are directly related to a Bayes minimum error estimate, while others are based on heuristic concept of class separability. These measures reflect statistical, geometrical and topological, or information-theoretic properties of data. In this paper, several measures from each category are evaluated. Expressions for class separability measures are presented in form of estimates based on a finite sample of population.

Usually, class separability is used as a criterion for dimension reduction to downscale the classification problem preserving data structure with inherent discriminatory information. The term feature selection is used in context of improved class discrimination. Application of class separability as a criterion in most cases leads to decrease in classification error rate, despite the exact relation to biases of different classifiers has not been established.

4.3.2 Parametric measures and their generalizations

Parametric measures make assumptions regarding underlying distribution in data, which is often unknown. To simplify the matters, probability estimates obtained from a data set can be used to substitute unknown parameters. However, such estimates may not be reliable (Pierson, 1998). Mahalanobis and Bhattacharyya distances are often used as Bayes minimum error estimates (Mao & Tang, 2011). As a class separability measure, Mahalanobis distance between classes can be compared to a sum of standard deviations of both classes. Mahalanobis distance increases with increasing distances between class centroids (means) and with decreasing within class variation (Everitt *et al.*, 2011). Mahalanobis distance assumes that covariance matrices of two classes are identical, their distribution is Gaussian, and prior probabilities of the classes are equal. When this is not at least approximately so, this measure is not meaningful and has to be substituted by one of the alternatives. Such an alternative is provided by a Normal Information Radius ($NIR_{A,B}$) (Everitt *et al.*, 2011). Given two sets of instances corresponding to classes A and B with mean vectors μ_A and μ_B, covariance matrices Σ_A and Σ_B, $NIR_{A,B}$ between the two classes is shown in Formula 34.

$$NIR_{A,B} = \frac{1}{2}\log_2\left(\frac{\left|\frac{1}{2}(\Sigma_A + \Sigma_B)\right| + \frac{1}{4}(\mu_A - \mu_B)^T(\mu_A - \mu_B)}{\sqrt{|\Sigma_A||\Sigma_B|}}\right) \tag{34}$$

Bhattacharyya distance ($a_{A,B}$) is a generalization of the Chernoff bound on Bayes error, which is widely used as a class separability measure (Fukunaga, 1990; Pierson, 1998). The Bhattacharyya coefficient expresses overlap between two statistical samples that correspond to two classes as shown in Formula 35.

$$a_{A,B} = \frac{1}{8}(\mu_A - \mu_B)^T \left(\frac{1}{2}(\Sigma_A + \Sigma_B)\right)^{-1} (\mu_A - \mu_B) + \frac{1}{2}\ln\left(\frac{\left|\frac{1}{2}(\Sigma_A + \Sigma_B)\right|}{\sqrt{|\Sigma_A||\Sigma_B|}}\right) \quad (35)$$

The first term represents class separability due to the means difference while the second term reflects class separability due to the covariance difference.

Measures that involve computation of covariance matrices and their inversion are computationally intensive. Another limitation associated with covariance matrices is that if the number of instances per class is less than the number of features, the covariance matrix is singular and inverse cannot be computed. Regularization is one possible solution to this problem.

4.3.3 Information-theoretic measures

Information-theoretic measures are based on examination of the concept of statistical independence between distributions of features and a class variable (Pierson, 1998).

Kullback-Leibler distance ($KL_{A,B}$) is an information-theoretic measure of relative entropy that measures discrepancy between two probability distributions (Kullback, 1968). If used as a class separability measure, it's a distance between histograms of features (Cantú-Paz, 2004). Kullback-Leibler distance is often used in filter feature selection, for example, in (Koller & Sahami, 1996). Symmetric version of this measure is given by Formula 36.

$$KL_{A,B} = \frac{1}{2}\left(P(\mathbf{x}|A)\ln\frac{P(\mathbf{x}|A)}{P(\mathbf{x}|B)} + P(\mathbf{x}|B)\ln\frac{P(\mathbf{x}|B)}{P(\mathbf{x}|A)}\right) \quad (36)$$

Mutual separability measure between two classes can be expressed by divergence ($DIV_{A,B}$), the measure derived from the Bayes rule, (Formula 37).

$$DIV_{A,B} = \big(P(\mathbf{x}|A) - P(\mathbf{x}|B)\big)\ln\frac{P(\mathbf{x}|A)}{P(\mathbf{x}|B)} \quad (37)$$

For multi-class problems, $P(\mathbf{x}|A)P(\mathbf{x}|B)DIV_{A,B}$ is accumulated for all pairs of classes. $DIV_{A,B} \geq 0$. Divergence is 0 for completely overlapped classes.

4.3.4 Proximity based and heuristic measures

Nonparametric measures are mostly based on density estimates and often utilize the neighborhood concept. Nonparametric estimates are used, for example, to assess multimodality, skewness, or any other structure in distributions of the data. Parzen and *k*-Nearest Neighbor (*k*-NN) based measures are representative in this category (Pierson, 1998). Both measures are based on defining the ratio of instances in a close vicinity of an instance in a

data set. These measures also provide a Bayes minimum error estimate and is used as class separability based feature selection, for example, in (Singh *et al*., 2002). The density estimation for instance **x** is defined by the ratio of neighbors k in volume v adjusted by the number of instance in a data set, M. Initial settings for v or k might significantly influence the result and require preliminary estimation. The density estimation for Parzen measure is measured with the volume of the local region fixed. For k-NN measure, the number of neighbors is fixed. The related approach is employed in adherence mapping (Ho *et al*., 2006) and used as a complexity measure called the fraction of maximum covering spheres. This approach is also used in density-based clustering such as DBSCAN.

Fisher's linear discriminant ($F_{A,B}$) does not make strong assumptions about normally distributed features, equal covariance matrices, and Gaussian classes contrary to linear discriminant analysis; it is a more flexible measure. For two classes A and B with the respective means (centroids) μ_A and μ_B, Fisher (Fisher, 1936) defined the separation between two distributions in feature j to be the ratio of the variance between the classes σ^2_{inter} to the variance within the classes σ^2_{intra}, (Formula 38).

$$F_{A,B} = \frac{\sigma^2_{inter}}{\sigma^2_{intra}} = \frac{(\mu_A - \mu_B)^2}{\sigma_A^2 + \sigma_B^2} \tag{38}$$

Not all features necessarily contribute to class discrimination. As long as one or a few discriminative features are found, the classes are linearly separable. If none of the features has a nonzero Fisher's linear discriminant ratio, it does not mean that the classes are not separable, but it means that the separating line is not parallel to the axis in the given feature space.

In discriminant analysis (Fukunaga, 1990), three scatter matrices are computed: intra-class (within class), inter-class (between class), and total scatter matrix. There are three popular scatter matrix based measures that use the idea of class means separation and covariance tightness (Fukunaga, 1990). An intra-class scatter matrix is similar to class covariance matrix, but it is adjusted with prior class probabilities, as shown in (Formula 39).

$$S_{intra,A} = \frac{M_A}{M}(\mathbf{x} - \mu_A)(\mathbf{x} - \mu_A)^T = P(A)\Sigma_A \tag{39}$$

Total intra class scatter matrix is accumulated over all classes. An inter class scatter matrix is computed as shown in (40) and accumulated over all classes.

$$S_{inter,A} = \frac{M_A}{M}(\mu_A - \mu)(\mu_A - \mu)^T \tag{40}$$

A total scatter matrix is a common covariance matrix, $S = \Sigma$. Average mean vector $\mu = P(A)\mu_A + P(B)\mu_B$.

Various scatter matrix based measures (Fukunaga, 1990; Pierson, 1998) use tuples S_1, S_2 from $\{S_{intra}, S_{inter}, S\}$, for example, as shown in (Formula 41) - (J_1) and (Formula 42) - (J_2).

$$J_1 = tr(S_{intra}^{-1} S_{inter}) \tag{41}$$

$$J_2 = \frac{tr(S_{intra})}{tr(S_{inter})} \tag{42}$$

A few other variants are listed in Pierson (1998). A class separability measure for feature selection based on the difference between intra-class distances and inter-class distances computed with respect to class centroids is used in Liang *et al*. (2008). Class centroids that correspond to the means of two classes, μ_A and μ_B, can be found in a different way with nonparametric estimates. The measures derived from scatter matrices include computation of intra-class and inter-class distances via average distances to a nearest neighbor from the same or different class respectively (Ho *et al*., 2006).

The ratio of average intra- and inter-class nearest neighbor distance $\partial_{A,B} = \frac{d_{same}}{d_{diff}}$ ($\partial_{A,B}$) is a class separability measure that is closely related to empirical measures and nonparametric estimates, such as scatter matrices. Intra- and inter-class ratio compares dispersion within class to the gap between classes involving a distance function. This measure has more flexibility than scatter-based measures as it conveys density and does not deal with class centroids, which can be a nontrivial task. $\partial_{A,B}$ is sensitive to outliers and can be misleading in case of uneven density and unusual class shapes. Heuristic measures based on scatter matrices are not directly related to Bayes minimum error estimate. Intra- and inter- class distances ratio is previously discussed in Subsection 3.2.3 with respect to decomposition based on local class separability.

4.3.5 Complexity measures

Geometrical complexity of a classification problem refers to length and shape of class boundaries, class distributions and/or density within classes, margins between classes, dimensionality, data size, overall presentation of a classification problem by a data sample (intrinsic unambiguity), and other factors directly or indirectly influencing Bayes minimum error and error of a particular classifier. Geometrical complexity measures examine class boundary and class margins explicitly, but characterize class separability indirectly. It can be useful in assessing class separability measures in aspects related to classification accuracy, but independent on a classifier choice (Ho, 2002; Ho *et al*., 2006).

Fraction of points on the class boundary (N1) is the number of instances connected to the opposite class by the edge of minimum spanning tree with

respect to all instances. It takes values in the interval (0..1), where smaller values correspond to better class separability. This measure is misleading when intra-class distance is smaller than inter-class distance, but classes are separable. For example, two elongated linearly separable classes with narrow margin between them.

The leave-one-out error rate of the 1-NN classifier (N2) is used as a complexity measure in (Ho, 2002) as it shows how proximity of instances from an opposite class (margin between classes) affects error rates of a basic distance-based classifier. The nonlinearity of the 1-NN classifier (N3) is a complexity measure that originates form the measure of nonlinearity of a linear classifier, L3, as described below.

Measure (T1) is based on the notion of adherence subsets used to describe the shape of class manifolds. An adherence subset is spherical ε-neighborhood with a radius ε centered on an instance. The neighborhood is expanded until it touches an instance from an opposite class. A list of such neighborhoods needed to cover two classes is a composite description of the shape of the classes (Ho *et al*., 2006). This is an interior description rather than a boundary description given by measure N1. The number and order of the retained adherence subsets indicate how much the instances tend to be clustered in hyperspheres or distributed in thinner structures (Ho *et al*., 2006). In a problem, where each instance is closer to instances of the other class than instances of the same class, each adherence subset is retained and is of a low order. When used as a measure of class separability, the count of the retained adherence subsets is normalized by the total number of instances. T1 values are in the interval (0..1], where smaller values correspond to better class separability.

The average number of points per dimension (T2) is a ratio of instances and features in a data set that is a rough indicator of data sparseness. T2 is a complimentary measure; it increases with elimination of irrelevant features.

Measures derived from error rates of a linear classifier (SVM with linear kernel trained with Sequential Minimum Optimization (SMO) are the minimized sum of the error distance of a linear classifier (L1) and the training error of a linear classifier (L2). These measures evaluate to what extent the classes are linearly separable. L1 is the sum of the differences in predicted and actual class, hence, a zero value indicates linear separability. L2 returns the training error for the same linear classifier.

The measure of nonlinearity of a linear classifier (L3) is an error rate from linear SVM classifier obtained from the test set that is created by linear interpolation from the original data. The test instance $\mathbf{x}$ is created from a pair of randomly selected instances of the same class $\mathbf{x}_1$ and $\mathbf{x}_2$ with random coefficients *rnd* in each feature *j*: $x^j = \text{rnd} * x_1^j + (1 - \text{rnd}) * x_2^j$. This measure is for the alignment of the decision boundary produced by linear SVM with the shape of the gap or overlap of class boundaries presented as convex hulls. Zero test error is an indication of good class separability. It is of particular interest in cases of uneven density and existence of subclasses. In multi-class problems the

results of L1, L2, and L3 are averaged over pairwise class decompositions. Measures of this category can be compared directly for different data sets.

4.4 Chapter summary

There are different approaches to estimate a discriminative power of the feature subset. In this thesis, only feature selection methods independent of a learning algorithm are considered in order to avoid inductive biases of those algorithms. In some methods, merits of individual features are evaluated one by one, while in other methods merits of different feature subsets are evaluated. "Myopic" feature merit measures disregard possible interactions between features evaluating each feature independently. Methods using "myopic" feature merit measures are successful in many cases, since the assumption about independence of features is often true in practice. "Non-myopic" measures take into account feature interactions considering values of the other features.

Estimating a quality of feature subsets is more computationally expensive, but can be advantageous, because even "non-myopic" measures cannot perceive all kinds of higher order relations that may exist between features. Some feature subset merit measures, for example, correlation-based feature subset merit measure, consider only particular feature interactions to find a subset, and therefore gain more computational efficiency.

Considering feature-feature, feature-class interactions and higher order relations between features is important for heterogeneous classification problems (Subsection 4.2.1). Different merit measures handle redundant and interacting features differently, because they are based on different assumptions about the data nature, for example, such as a tendency to group by classes in distance-based methods. In such a way, individual feature / feature subset merit measures may correspond to the underlying data characteristics of different data set to a various extent.

The Information gain feature merit measure is based on the mutual information and estimates a general dependence of random variables without making any assumptions about the nature of their underlying relationships.

ReliefF assumes that a good feature should differentiate between instances from different classes and should have the same value for instances from the same class. ReliefF is able to identify interactive features, but tends to consider redundant features as important.

The rationale for a correlation-based subset selection is the following. A good feature subset is one that contains features highly correlated with (predictive of) a class variable, yet uncorrelated with (not predictive of) each other.

In this thesis the considered individual feature merit measures producing rank of features are used as a part of IPA to evaluate dissimilarity between subproblems.

5 BIDIRECTIONAL DATA PARTITIONING

This chapter presents a novel Bidirectional Data Partitioning (BDP) technique that follows a decomposition approach for a general feature space heterogeneity proposed in Chapter 4. This decomposition approach does not assume existence of contextual features that facilitate data partitioning or that homogeneous regions can be found among pairwise or one-against-all class combinations. The main challenge of finding regions as sets of instances with associated sets of relevant features, is to encompass search in the space of instances and the space of features simultaneously. In BDP, this task is accomplished via adaptive learning of distance metric with feature weighting. Weighted distances are given as inputs to a distance-based clustering algorithm. Because clustering cannot obtain regions with increased class separability directly, agglomerative merging of clusters is performed comparing feature weight profiles in clusters. Merged groups of instances represent subproblems of the original classification problem. Relevant subset of features is found in each subproblems by means of feature weights or an external feature selection technique. Subproblems are modeled separately. Selection of the right model for novel instances is a distance-based procedure. A few alternative solutions presented in implementation of BDP are discussed.

Implementation of the proposed approach adopts a local neighborhood search to adjust feature weights iteratively and then involves a weighted distance-based grouping procedure. Feature weights reflect local improvement of class discrimination; specifically, they promote increased density of the same-class instances while separating instances from a different class. Better class separability reflects wider margin between classes and larger distance between class centroids, less complex form of boundary between classes, and

higher density within classes. A criterion based on intra- and inter-class distances is presented among nonparametric class separability measures in Section 5.1. Bidirectional data partitioning is presented as an optimization task in Section 5.2. A practical solution to the optimization task that utilizes a local neighborhood concept is described in Section 5.3. Bidirectional partitioning as a multi-model approach is presented under ensemble framework in Section 5.4. Additional aspects and related approaches are covered. Summary and conclusions are provided in Section 5.5.

5.1 Criterion function

In classification tasks, or supervised learning, data instances are classified according to certain observable characteristics. In clustering, or unsupervised learning, and semi-supervised learning, the same set of observable characteristics is available, except for the class labels, which can be partially available. There are measures used in clustering, filter feature selection, and data complexity evaluation that make suitable candidates for decomposition evaluation. Those measures bring their inherent bias and some of them are not directly related to minimum Bayes error rate, but they are proven to be effective in classification tasks (Fukunaga, 1990; Ho *et al*., 2006).

Data structure can be seen beyond class labels at different levels of granularity (subclasses and super classes). In different projections of the feature space, unstable feature relevance shows off in form of uneven density regions and complex nonlinear class boundaries (illustrated by synthetic data examples presented in Chapter 6 and Appendix 3). Criterion for decomposition of a classification problem onto a few simpler subproblems then can be derived from complexity characteristics, and in particular, class separability measures.

Class discrimination can be improved via improved class separability. This statement is widely used in the literature on developing feature selection methods based on class separability, distance functions used in classification, clustering for classification, and classification itself. The approach proposed here increases densities of classes increasing margins between classes. The next section reviews and summarizes several parametric and nonparametric class separability measures in order to provide a background for further development of BDP technique.

5.1.1 Criteria based on class separability

Criteria based on class separability have been extensively used in filter feature selection (Liu & Motoda, 1998). A problem related to unstable feature relevance in classification has recently received increased attention in clustering. Subspace clustering, also known as two-mode partitioning and block clustering, is an extension of traditional clustering that seeks clusters in different feature subspaces. A search is performed simultaneously in the subspace of instances

and in the subspace of features in order to partition data. In other words, both rows and columns of a data table are assigned to one or more clusters. Elements in the same cluster are close to each other in terms of a pre-defined distance function or a similarity measure.

The rationale for choosing a criterion for bidirectional data partitioning in classification is based on the following observation. Classes are groups of conceptually meaningful objects that share common characteristics. Clusters are similar by nature. In clustering, an evaluation function is based on intrinsic properties of data, a criterion that reflects data structure with respect to a chosen distance function or a similarity measure. Evaluation functions used in clustering are often applied as class separability measures in case of known class labels, for example, in filter feature selection and feature extraction. These measures provide additional description of labeled data indirectly related to error rates from a particular classifier. Improved class separability in many cases implies easier class discrimination in terms of classification rule complexity, resistance to overfitting, and accuracy of prediction. Examples of class separability measures include a variety of scatter matrix based, Fisher's linear discriminant ratio based, and Bhattacharyya distance-based measures among others (Fukunaga, 1990; Pierson, 1998).

Intuitively, in order to improve class discrimination in high-dimensional data one may find a subset of features such that data projection onto corresponding dimensions will produce smaller inter-class distance and larger intra-class distance and wider margin between classes. This reasoning comes from various class separability measures and minimum Bayes error estimates. Related approach is used in subspace clustering (Parsons *et al.*, 2004). Irrelevant dimensions hide clusters in noisy data.

In order to pick suitable class separability based criterion for decomposition, parametric and nonparametric measures has been analyzed in Section 4.3 and preliminary experiments are carried in Section 6.1 in order to relate geometrical properties of data and class separability measures. It has been shown that intra- and intra-class based measure $\partial_{A,B}$ (Subsections 3.2.3 and 4.3.4) is generally applicable, but computationally demanding. The next best candidate is a scatter matrix based measure that is a ratio of traces over intra-class and inter-class scatter matrices, which is basically the ratio of respective variances accumulated over all features. Scatter matrix based class separability requires much less computations. Experiments with these measures in presence of irrelevant features suggest that reduction of intra-class distance, or variance within class for each class separately will result in increased inter-class distance, or variance with respect to the mean vector obtained disregarding class labels.

A class separability measure chosen for BDP is a function of within and between class distances. There are variants of this measure that compute intra-class difference via class means (Liang *et al.*, 2008) and as an averaged distance to the nearest neighbor of the same class (Ho & Basu, 2002). In our implementation we use a synergy of these two considering the fact that class

means are not meaningful if clusters are non-globular and have an odd shapes, density, or boundaries. In order to obtain an analytical solution, a criterion function based on a class separability measure must be differentiable function of distance with weights assigned to features and instances likewise.

5.1.2 Criterion based on intra- and inter-class distances

Let us denote $\omega_j^l,\ j = 1 \dots N,$ the criterion component to optimize in feature f_j over a subset (group) of instances l, $l = 1 \dots L,$ on the training set TR $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_M, y_M)\}$ of size $M,$ which includes D classes, $d = 1 \dots D$. Class d includes M_d instances. Group l includes M_l instances of the training set. If the group is a part of class, it's denoted M_d^l. The number of instances of class d in group l is denoted as $M_{l,d}$. The criterion component ω_l^j includes calculation of a distance between two instances r and s in feature f_j, $d_{r,s}^j$, which is assigned a weight $w_{r,s}^j$ (pairwise weight). The optimal weighting will promote optimization of class separability inside a group of instances (a homogeneous region), $\omega_l = \sum_{j=1}^N \omega_l^j$. We need to obtain optimal grouping with minimum ω_l inside them, therefore the criterion is $\sum_{l=1}^L \sum_{j=1}^N \omega_l^j$.

Numerous distance functions are suggested in the literature for distance measures on individual features. Some of them are considered in Appendix 2.2. Particular choices of a distance function reflect the analysis goal.

Manhattan distance is more preferable than Euclidian distance metric for high-dimensional applications (Aggarval *et al.*, 2001), therefore the distance used throughout BDP is a combination of normalized Manhattan distance (divided by the range of feature's values) for continuous features and Value Difference Metric (VDM) (Stanfill & Waltz, 1986) for nominal features.

A non-normalized Manhattan distance in a continuous numeric feature is $\delta_{r,s}^j = |x_{r,j} - x_{s,j}|$. In case of discrete numeric, boolean, or symbolic features it takes form $\delta_{r,s}^j = \begin{cases} 0, \text{if } x_{r,j} = x_{s,j} \\ 1, \text{if } x_{r,j} \neq x_{s,j} \end{cases}$. Normalization (Formula 43) prevents contribution of features that take values on a larger interval to be overrated. For example, in a similar distance function in COSA subspace clustering technique the distance between instances in one dimension is scaled by the average distance between instances in that dimension (Friedman & Meulman, 2002). This is a scale for measuring "closeness" in each dimension, for which another measure of dispersion can be used. A reason to use variance in preference to other dispersion measures is that variance of the sum (or the difference) of uncorrelated random variables (independent features) is the sum of their variances. Based on our experiments with class separability measures in Section 6.1, scatter matrices based measure J_2 that used variances (mean involved) has performed almost as good as intra- and inter-class distance-based measure.

$$d_{r,s}^j = \frac{\delta_{r,s}^j}{|\max_j - \min_j|} \tag{43}$$

In case, an equal influence of all features in distance computation is preferable when weights $w_{r,s}^j$ are equal, the average distance s^j computed over all pairs of instances in TR should be used instead of $|\max_j - \min_j|$ in Formula 43, same as in COSA. This approach is more advanced, but requires additional computations making the technique more computationally expensive.

A weighted distance $D_{r,s}$ in all features is defined by Formula 44, where the weights $w_{r,s}^j$ sum up to 1 over all features.

$$D_{r,s} = \sum_{j=1}^{N} w_{r,s}^j d_{r,s}^j \tag{44}$$

If two well-separated classes are elongated and the margin between them is narrow (most instances connected to the instances of a different class by the edge of Minimum Spanning Tree), the difference between intra- and inter-class distances will be relatively small. In a criterion function it can be adjusted by a parameter β so that the difference within a partition l is $d_{l,same}^j - \beta d_{l,diff}^j$. The difference criterion function derives a function of weights in a less complicated form than the ratio criterion function, as will be seen from the subsequent expressions.

In order to evaluate contribution of individual features in ω_l^j we have to assume their independent contribution to class discrimination. This assumption does not hold in practice in many cases, but nevertheless successfully used in many data mining techniques. ω_l^j can be presented via $d_{r,s}^j$ as a distance from the instance r to be evaluated to the neighboring instance s of the same and different class(es). A class-membership function is given by (Formula 45).

$$g(r,s) = \begin{cases} 1, & \text{if } x_r \text{ and } x_s \text{ are in the same class} \\ 0, & \text{otherwise} \end{cases} \tag{45}$$

The intra-class distance $d_{l,same,d}^j$, which is calculated over the instances of group l in feature f_j, is an averaged distance to the nearest neighbor of the same class (Formula 46), where $1NN(r,s)$ is defined in Formula 47.

$$d_{l,same,d}^j = \frac{1}{kM_{l,d}} \sum_{r=1}^{M_{l,d}} \sum_{s=1}^{M_{l,d}} 1NN(r,s) g(r,s)\, w_{r,s}^j d_{r,s}^j \tag{46}$$

$$1NN(r,s) = \begin{cases} 1, & \text{if } x_r \text{ is a nearest neighbor of } x_s \\ 0, & \text{otherwise} \end{cases} \tag{47}$$

The intra-class distance $d_{l,same}^j$ calculated over the instances of group l in feature f_j is an average distance to the nearest neighbor of different class (Formula 48).

$$d_{l,diff,d}^{j} = \frac{1}{kM_{l,d}} \sum_{r=1}^{M_{l,d}} \sum_{s=1}^{M_l - M_{l,d}} 1NN(r,s)\big(1 - g(r,s)\big)\, w_{r,s}^{j} d_{r,s}^{j} \tag{48}$$

The criterion constituent based on an intra- and inter-class difference is given by Formula 49, where D_l is a number of classes included in group l and $\beta > 1$.

$$\omega_l^j = \frac{1}{D_l} \sum_{d=1}^{D_l} d_{l,same,d}^{j} - \beta d_{l,diff,d}^{j} \tag{49}$$

An accumulated criterion that minimizes a distance in each dimension can be conveniently applied in case of L^1 norm, or Manhattan distance, in order to minimize distance used in a corresponding class separability measure. A similar criterion that ignores class labels is used in subspace clustering (Friedman & Meulman, 2002). A similar difference criterion is used for feature selection in Liang *et al*. (2008). The difference criterion function allows presenting the weight function in a less complicated form than the ratio criterion function in order to obtain an analytical solution. However, for the sake of simplicity in practical implementation $-\beta d_{l,diff,d}^{j}$ can be omitted in Formula 49. This option is available in BDP as *clustering inside classes* (CIC).

If $\left|\omega_l^j\right|$ was taken in Formula 50, the criterion would have to be maximized, same as the class separability. In Formula 49 the difference is negative in case of good class separability and ω_l^j is minimized, which is convenient if to think of it as a distance component.

As an alternative to the intra- and inter-class difference in one feature, (Formula 49), a simplified criterion can be used based on measure of feature values overlap within group. That will lead to a suboptimal solution, but can be used as a substitute considering drastic reduction in computation it entails (Skrypnyk, 2008). Another alternative is the use a ratio of within class variance and total variance computed inside group in one feature. This measure is a heuristic derived from a scatter matrix based class separability measure (Formula 42), though accumulated in all features, it will not be equal to the original measure, and also leads to a suboptimal solution.

To some extent, counterparts of most class separability measures that are transformed to a sum of individual feature components can be used in criterion function. Examples are Kullback-Leibler, Fisher linear discriminant, etc. Criteria not based on distance may not be directly linked to distance-based clustering, but other type of clustering can be used, for example, information-theoretic clustering. All of these alternatives are worth exploring and comparing in the future research.

5.2 Solution to the optimization task

In this section the task of bidirectional data partitioning is formalized. Data partitioning is performed in a metric space with a distance function involved. The distances are iteratively transformed by feature weights until the process stabilizes. The constituents of bidirectional data partitioning are a distance-based clustering to obtain subgroups, a cosine similarity function for agglomerative merging subgroups to form groups, feature selection by means of feature weights, feature selection by external criteria, and distance-based selection of a local model in ensemble. In our implementation, the criterion is reduced to minimization of within class distances following empirical observation from preliminary studies. In this form it resembles the criterion used in COSA subspace clustering (Friedman & Meulman, 2002). Description and notations are intentionally chosen to connect with COSA to in order to relate distance-based feature space heterogeneity decomposition in classification and distance-based subspace clustering.

5.2.1 Grouping instances in feature subspaces

The goal is to partition the training set *TR* $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_M, y_M)\}$ of size *M*, where $\mathbf{x}_i$ are vectors of the form $\langle x_{i,1}, \dots, x_{i,j}, \dots, x_{i,N} \rangle$ and $x_{i,j}$ are feature values of $\mathbf{x}_i$, onto *groups* according to the specified encoder c^* (Formula 50). On an upper level, "encoder" is a function of *TR* implementing a distance-based clustering algorithm with its pre-set parameters for a distance-based class separability. The relative influence of each feature f_j in a distance component is regulated by the corresponding weight w^j. Optimal weighting $w = \{x^{f_j}\}_1^N$ has to be found as a part of the grouping process jointly minimizing the criterion with respect to encoder c and weights w in order to obtain a solution (c^*, w^*).

$$(c^*, w^*) = \arg\min_{(c,w)} Q(c, w) \tag{50}$$

In COSA criterion, a ratio of averaged weighted distant components $d_{r,s}^k$ computed over all pairs of instances within a cluster l and computed over all instance pairs on *TR* is considered. Then an average obtained in all features is taken to optimize. In bidirectional partitioning, the difference $d_{l,same,d}^j - \beta d_{l,diff,d}^j$ is taken instead of $d_{r,s}^k$. This magnitude can be perceived as a "balanced" distance component.

In order to obtain an analytical solution an entropy-based regularization can be used. We choose the same regularization term as in COSA, $\lambda w_{r,s}^j \ln w_{r,s}^j$ that shapes the criterion function, where parameter λ controls the degree of deformation caused by regularization. A practical value for this parameter is $\lambda = 0.2 \dots 0.5$ (Friedman & Meulman, 2002), and k = 1. This regularization term is added to (46) and (49) deriving (51 and 52). This regularization term is

required to distribute high weights among feature subspaces, where the number of features in the subspace is regulated by λ. Increasing its value will encourage higher dimensionality subspaces. $\lambda = \infty$ forces to distribute weight equally among all features in the group. Without this regularization term, $\lambda = 0$ an analytical solution would give a maximal weight to the most discriminative feature in terms of the class separability criterion used, ignoring the rest of features.

$$d^j_{l,same,d} = \frac{1}{kM_{l,d}} \sum_{r=1}^{M_{l,d}} \sum_{s=1}^{M_{l,d}} kNN(r,s) g(r,s) \left(w^j_{r,s} d^j_{r,s} + \lambda w^j_{r,s} \ln w^j_{r,s} \right) \quad (51)$$

$$d^j_{l,diff,d} = \frac{1}{kM_{l,d}} \sum_{r=1}^{M_{l,d}} \sum_{s=1}^{M_l - M_{l,d}} kNN(r,s) \left(1 - g(r,s)\right) \left(w^j_{r,s} d^j_{r,s} + \lambda w^j_{r,s} \ln w^j_{r,s} \right) \quad (52)$$

β is a parameter used to control the impact of the inter-class distance, a suggested value is $\beta = 2$ (Liang *et al.*, 2008). Our empirical observations with synthetic data have shown that for cases with $\beta \geq 2$ class separability is reached (Section 6.2).

Taking into account two facts, that $kNN(r,s)$ and $g(r,s)$ are "membership" functions taking values either 0 or 1, and that for Manhattan distance $\sum_{r=1}^{M} \sum_{s=1}^{M} \sum_{j=1}^{N} (\cdot) = \sum_{j=1}^{N} \sum_{r=1}^{M} \sum_{s=1}^{M} (\cdot)$, we can define a criterion as shown in Formula 53 where ω^j_l is given by Formula 49.

$$Q(c, \{w_l\}_1^L) = \sum_{l=1}^{L} W_l \left(\sum_{j=1}^{N} \left(w^j_{r,s} \omega^j_l + \lambda w^j_{r,s} \ln w^j_{r,s} \right) + \lambda \ln N \right) \quad (53)$$

In (53) $W_l = M_l^2$ gives equal weight to all solution groups or can control the number of instances in a group if taken as a function of the number of instances in a group.

Minimization of the criterion $Q(c, \{w_l\}_1^L)$ that uses weights assigned to features inside each group will encourage solutions with one "best" feature selected for the group. Since the goal is to find a subset of features, a function that achieves its minimum value for equal weights and grows as the weights become more unequal is needed for regularization. The negative entropy function (Formula 54) satisfies this condition.

$$e(w_l) = \sum_{j=1}^{N} w^j_l \ln w^j_l \quad (54)$$

The negative entropy function is illustrated for the two-dimensional case (feature weights w^1, w^2) in Figure 9.

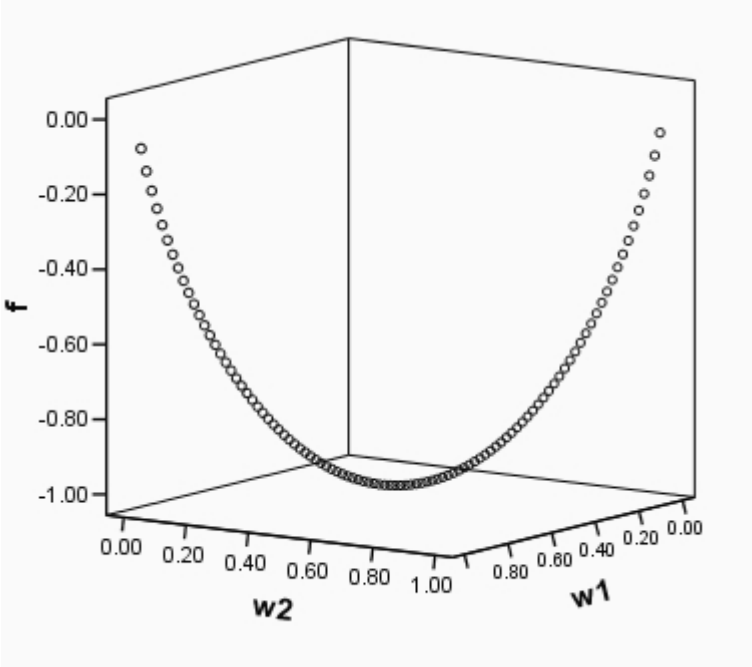

FIGURE 9 Simulated graphical presentation of a negative entropy function, $f(w^1, w^2) = \sum_{j=1}^{2} w^j \ln w^j$, $w^1 + w^2 = 1$. Weights (w^1, w^2) take values (0.01, 0.99),…,(0.5, 0.5),…,(0.99, 0.01). In bidirectional partitioning, weights are initially equal, so in this case weights would take values (0.5, 0.5),…,(0.99, 0.01).

By analogy with COSA, we introduce a distance controlled by λ, which is a part of Formula 55.

$$D_{r,s}^{(\lambda)} = \sum_{j=1}^{N} \left(w_{r,s}^{j} \omega_{l}^{j} + \lambda w_{r,s}^{j} \ln w_{r,s}^{j} \right) + \lambda \ln N \tag{55}$$

In Formula 56 term $\lambda \ln N$ is added in order to provide a translation so that overall distance is 0 when the component distances in f_j are all 0, that is $\min_{\mathrm{w}_l} D_{r,s}^{(\lambda)}[\mathrm{w}_l] = 0$ whenever $\left\{ d_{r,s}^{j} = 0 \right\}_{j=1}^{N}$.

In general, the rule of assigning a unique weight to a each feature with respect to a particular group will be the following. *Inside the group l, the weight w_l^j in feature f_j should be proportional to the average distance between instances of the same class balanced with the average distance between instances of different classes, $w_l^j \sim \omega_l^j$.*

Now the solution for the criterion function defined in Formula 53 can be defined with respect to the group weights vectors, as shown in Formula 56.

$$(c^*, \{w_l^*\}_1^L) = \underset{(c, \{\mathrm{w}_l\}_1^L)}{\arg\min}\, Q(c, \{\mathrm{w}_l\}_1^L) \tag{56}$$

Here Q is a function of parameters $\{\mathrm{w}_l\}_1^L$. In order to find weights values that minimize the criterion function over all possible groupings, an equation for parameters has to be solved. In this case, the criterion function is differentiable;

hence, unknown weights can be expressed explicitly in analytic form. The criterion function argument w_l^j at the extremum point can be obtained from Formulae 49, 51, 52, and 53 by the first derivative (Formula 57). This weight is related to a feature f_j with respect to a group of instances l.

$$w_l^j \sim \exp\left(-\frac{\lambda + \omega_l^j}{\lambda}\right) \tag{57}$$

In order to provide $\sum_{j=1}^{N} w_l^j = 1,\ l = 1 \ldots L,$ the expression in Formula 57 is divided by total weight, as shown in Formula 58.

$$w_l^j = \frac{\exp\left(-\frac{\lambda + \omega_l^j}{\lambda}\right)}{\sum_{p=1}^{N} \exp\left(-\frac{\lambda + \omega_l^p}{\lambda}\right)} \tag{58}$$

As intra-class and inter-class distances in feature f_j change towards better class separability, the weight assigned to feature w_l^j increases to encourage usage of this feature for group l. Figure 10 shows the weight function with $\lambda = 0.5$.

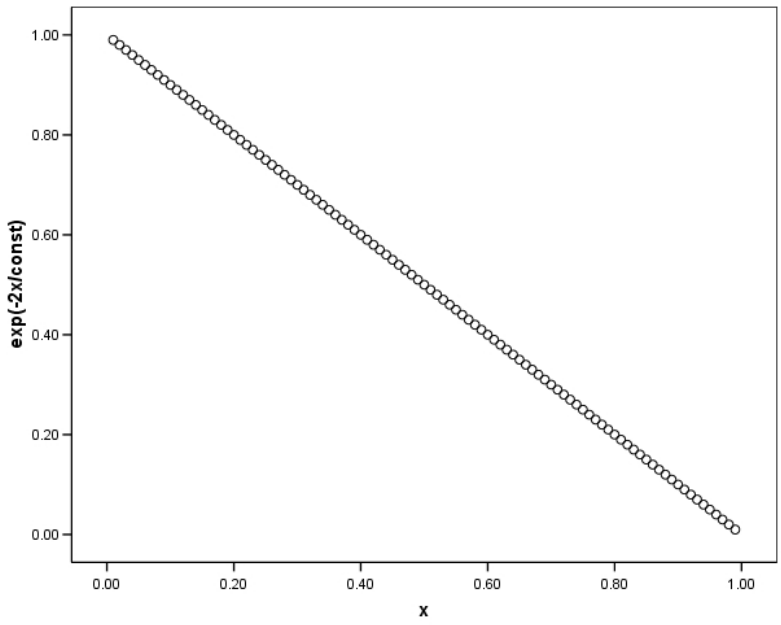

FIGURE 10 Simulated graphical presentation of $w_l^j(\omega_l^j) = \exp(-2\omega_l^j/\text{const})$, $\lambda = 0.5$, $N = 3$, $\sum_{j=1}^{N} w_l^j = 1$. Shown for discriminative features only, $0 \le \omega_l^j \le 1$.

In classification and regression, regularization technique is used to alleviate or prevent the overfitting problem (Mao & Tang, 2011). Regularization is often performed by introduction of extra terms, for example, penalty of complexity to optimization function. Thus, regularized solution is usually suboptimal with respect to the original optimization function.

A theoretical part of bidirectional partitioning has a lot in common with COSA, therefore plenty of details can be found in (Friedman & Meulman, 2002). In particular, considerations related to parameter λ. Optimization strategy for BDP described in the next section can be conveniently adopted from COSA, though alternatives are a subject of future research.

5.2.2 Optimization strategy

The reasoning in obtaining weight function for a feature in a particular instance (single weight) w_r^j follows the same logic as for COSA's weight function. A similar search strategy based on a local neighborhood is also applied here. For a classification task this search strategy will undergo some transformations that will be discussed along with other implementation details in the next section.

Assuming that suitable criterions function $Q(\cdot)$ has been chosen to evaluate relevance of feature subsets over different data partitions (groups of instances), the optimization task is reduced to a search performed simultaneously in the feature space and instance space. As a result, class separability in every partition should be better than in the original data space. An exhaustive search is computationally prohibitive in most realistic problems. Therefore, one needs to explore a search strategy for suboptimal results. Literature in subspace clustering suggests many search strategies for bidirectional (two-mode) data partitioning (Rosmalen *et al.*, 2009). Among the best search strategies is a heuristic search strategy based on the neighborhood concept. In particular, this search strategy has been successfully applied in COSA (Friedman & Meulman, 2002). Practical implementation of COSA search strategy is related to the criterion function. The criterion used in BDP is related to that used in COSA, therefore heuristic search can be performed in a similar way.

Most straightforward way to find subspaces of features is to project the instances in all possible subspaces and find those with better class separability. Such an approach, however, is not feasible, because complete enumeration of the candidates is impossible. A particular search strategy is used to obtain a local minimum of the criterion function. The strategy is to explore local neighborhood of an instance in order to adjust single weight assuming that all instances in this neighborhood belong to the same group. This local neighborhood is found using a certain distance function and pairwise weights assigned to a distance between two instances in a particular feature. In (Friedman & Meulman, 2002) the weighted inverse exponential distance is obtained from the regularized criterion function with a substitution using *f*-mean. In our case, an averaged difference of distances from the same and different class ω_l^j can be considered as a measure of dispersion of distance differences in one feature, in analogue to a measure of dispersion of distances in (Friedman & Meulman, 2002). After application of *f*-mean the criterion changes as shown in Formula 59. This criterion depends only on the encoder.

$$Q(c) = \sum_{l=1}^{L} \frac{1}{N} \sum_{j=1}^{N} -\lambda \exp\left(-\frac{\lambda + \omega_l^j}{\lambda}\right) \tag{59}$$

Then, the quantity to be optimized, which is essentially the difference of distances from the same and different class, can be considered as a *discriminating weighted exponential distance*. It brings closer instances from the same class while separating instances from different classes in local regions. Discriminating weighted exponential distance (60) can be viewed as a homogeneity measure of the instance space. More theoretical details in obtaining this type of distance can be found in (Friedman & Meulman, 2002).

$$D_{r,s}^{(\lambda)} = -\lambda \ln\left(\sum_{j=1}^{N} w_{r,s}^j \exp\left(-\frac{\lambda + \omega_r^j}{\lambda}\right)\right) \tag{60}$$

In (60) ω_r^j is an approximation of ω_l^j made using local neighborhood, as shown in (61).

$$\omega_r^j = \frac{1}{k} \sum\nolimits_{s \in kNN(r)} d_{r,s}^j g(r,s) - \beta d_{r,s}^j \big(1 - g(r,s)\big) \tag{61}$$

As the constraints of known groups L are not present in the distance given by Formula 60, weights $\widetilde{w}_{r,s}^j$ are the pairwise weights that have no group constraints, as shown in Formula 61.

$$\widetilde{w}_{r,s}^j = \frac{\exp\left(-\frac{\left[d_{r,s}^j g(r,s) - \beta d_{r,s}^j \big(1 - g(r,s)\big)\right] + \lambda}{\lambda}\right)}{\sum_{p=1}^{P} \exp\left(-\frac{\left[d_{r,s}^p g(r,s) - \beta d_{r,s}^p \big(1 - g(r,s)\big)\right] + \lambda}{\lambda}\right)} \tag{62}$$

Pairwise weights given by Formula 62 are not the same for all same-class or different-class instance pairs in a local neighborhood, but the set of features for which pairwise weights are large for all instances in the neighborhood tends to be a super set of those for which actual solution weights should be large. This statement is illustrated on a synthetic example later in this subsection.

For a clustering task in (Friedman & Meulman, 2002) intersections of highly weighted feature subsets in two different groups are not allowed. Thus, single weight w_r^j (Formula 63) substitutes $\widetilde{w}_{r,s}^j$, in order to calculate pairwise weight $w_{r,s}^j$, which would be calculated as a maximum of single weights in each feature adjusted by the sum of weights in all features (Formula 64).

$$w_r^j = \frac{\exp\left(-\frac{\lambda + \omega_r^j}{\lambda}\right)}{\sum_{p=1}^{P} \exp\left(-\frac{\lambda + \omega_r^p}{\lambda}\right)} \tag{63}$$

$$w_{r,s}^{j} = \frac{\max\left(w_r^j, w_s^j\right)}{\sum_{p=1}^{P} \max\left(w_r^p, w_s^p\right)} \tag{64}$$

In order to distinguish all different weights mentioned so far, further we assume that single weights w_r^j are found by the search procedure in order to approximate weights in the actual groups w_l^j, while pairwise weights $w_{r,s}^j$ are needed in distance calculation $D_{r,s}^{(\lambda)}$ in order to find a local neighborhood of an instance.

By analogy with COSA, the number of neighbors can be chosen as $k = \sqrt{M}$. The value of parameter k can be adjusted considering the number of ungrouped instances after several experimental trials (Friedman & Meulman, 2002).

A practical solution for the optimization task entails some adjustments to the original expressions. The next section discusses details and algorithmic steps of BDP.

5.3 Description of the Bidirectional Data Partitioning technique

This chapter describes details of Bidirectional Data Partitioning (BDP) technique. These details cover important elaborations on weight adaptation, procedure of merging subgroups, and integration of classifiers.

5.3.1 Weights adaptation

Application of a local neighborhood search for weights adaptation assumes that majority or all neighbors belong to the same group. In practice, this assumption can only be verified on a benchmark data set with known group membership for all instances. Preliminary experiments have shown that instances from a different group often appear among *k* neighbors. This will be further referred to as error I. In order to explain error I, one has to consider mean and standard deviation in values of relevant and irrelevant features. Class conditional distribution of feature values suggests a non-zero probability of appearance of the different-group instances in the neighborhood due to irrelevant features. Small ω_l^j requires all intra-class distances to be small and all inter-class distances to be large, whereas irrelevant features contribute to these distances calculation. Irrelevant features can be assigned higher weights. It is expected that there will be more highly weighted features than there should be, but all relevant features will appear among them.

Highly weighted irrelevant features cause intersection of distance interval for instances of the same class in the same group and distance interval for instances of the same class in different groups. This will be further referred to as error II. As a result, DBSCAN would assign all instances of the same class to one group, which would be wrong.

In order to remove this negative effect, pairwise weights (Formula 64) should be calculated differently. If highly weighted features mostly coincide for two instances, minimum of feature weights is used in (Formula 64) instead of maximum, as these two instances are likely belong to the same group, and distance between them should be decreased using smaller weights in every feature, disregarding their class membership.

We apply the Importance Profile Angle (*IPA*) measure (Formula 65) for weight vectors in order to determine the extent to which profiles of feature importance differ. $IPA = 0.5$ is used as a default threshold. If $IPA \in (0, 0.5]$, profiles of feature weights mostly coincide; if $IPA \in (0.5, 1)$ – vice versa.

$$IPA_{r,s} = \frac{2}{\pi}\arccos\left(\frac{\sum_{j=1}^{N} w_r^j w_s^j}{\left(\sum_{j=1}^{N}\left(w_r^j\right)^2\right)^{\frac{1}{2}}\left(\sum_{j=1}^{N}\left(w_s^j\right)^2\right)^{\frac{1}{2}}}\right) \quad (65)$$

The preliminary experiments have shown that error I is greater than error II, and correction of error II only does not eliminate the problem. With application of IPA-guided pairwise weight calculation distance discrepancy was greatly improved according to preliminary experiments.

In order to avoid group intersections, it has been recommended that subsets of highly weighted features in different groups should not intersect (Friedman & Meulman, 2002). It is suggested to take maximal of single weights in pairwise weight computation (Formula 64). However, in classification tasks globally relevant features are common, and they should not be ignored. During single weight computations in a local neighborhood some instances may appear with equally distributed feature weights. Such instances should be assigned to a separate group at this stage and removed from the local neighborhood.

5.3.1 Merging subgroups and feature selection

A grouping procedure takes instances with assigned weights in every feature as an input. At the output it produces group labels for every instance. As the goal of weighting is non-intersecting or minimally intersecting distance intervals for intra-class distances at different groups, a grouping procedure succeeds if weighting succeeded.

Preliminary experiments have shown that distance-based grouping procedures (DBSCAN) was able to identify components of the groups, which mostly contain instances of the same class. However, without IPA-guided pairwise weight calculation all same-class instances are assigned to the same group, which was incorrect. After clustering, subgroups are joined according to feature weights profiles (Formula 65). We have used a cut-value for IPA to establish which subgroups should be merged.

Finally, group descriptions are obtained as enumeration of instances from the training set and assigned feature weights. In order to build a predictive a model for each group, feature selection should be performed by means of

feature weighting, or feature weights should be given directly to some distance-based learning model, such as *k*-Nearest Neighbor.

Feature selection can be performed by translation of continuous weights distributed on the interval [0…1] into binary weights {0,1}. It can be also performed establishing the weight cut-off value computed as a median for a ranked weight set. Features with weights above the cut-off values are retained.

However, feature selection by means of feature weighting can be incorrect. For example, if a feature has been assigned a high weight, it means that in a subgroup consisting of one class this feature has high probability of a particular value(s). At the same time, this feature may be assigned high weight in another subgroup consisting of different class, due to high probability of the same value(s). This feature is not discriminative and should be penalized.

Therefore, feature selection can be performed according to an external evaluation function in addition or instead of feature selection by means of feature weighting. We have implemented a possibility to use any feature selection technique available in WEKA open source software (Hall *et al*., 2009).

5.3.2 Description of local regions and ensemble construction

As the group memberships are established for all training instances and models are built for every group, the next step is to select an appropriate model for unclassified instances. In order to obtain a group membership for a new instance $(x_q, ?)$, where ? denotes the unknown class label, feature filters of every group $l = 1 \ldots L$ should be subsequently applied to this instance and the distance between new instance and all instances in every group should be found according to Formula 66. In order to minimize possible error associated with the cut-off value for feature selection, we have used group weights as pairwise weights in calculation of Formula 66. The necessity to do this adjustment has been demonstrated in Skrypnyk (2008).

$$D_{q,l} = \frac{1}{M_l} \sum_{i=1}^{M_l} \sum_{j=1}^{N} d_{i,q}^{j} \tag{66}$$

Group membership for the new instance is then defined by Formula 67.

$$y_q = \min\left(D_{q,l}\right) \tag{67}$$

After the group membership has been identified, a classifier associated with the selected group should be applied to a new instance. Among the candidate groups, the one with globally relevant features should be considered, while the noise group should be ignored.

5.3.3 Implementation

In practice, the bidirectional partitioning part is accomplished via two steps: (1) weighted distance-based grouping of instances, where distance is computed using *pairwise* weights computed based on single weights, and (2) feature

selection for each group of instances based on feature weights or other feature selection technique. Thereafter, the component classifiers are built on subsets of instances and associated subsets of features. Integration of component classifiers is performed in accordance with a chosen integration technique to perform a prediction for test instances.

Therefore, the main five steps of BDP technique include:

- features weights adaptation;
- weighted distance-based grouping of instances;
- feature selection for a group of instances;
- building local models – the component classifiers;
- integration of the component classifiers.

We have created a flexible experimental environment with extended possibilities for research beyond what is covered in this study. The BDP technique is implemented as a part of WEKA-3-6 open-source software (Hall *et al.*, 2009; Witten & Frank, 2005) as a meta-classifier. It may use any classification technique available in WEKA as a base classifier, including other meta-classifiers, such as boosting, multi-class classifiers, and classifiers with embedded feature selection technique.

Weights adaptation procedure recursively updates two type of weights: (1) feature weight for an instance called *single weight*, $\overline{w}_i^{f_j}$, and (2) feature weight for a pair of instances called *pairwise* weight, $\overline{w}_{r,s}^{f_j}$.

The process begins with equal pairwise weights used to calculate distances $\overline{D}_{r,s}^{f_j}$ needed to find k nearest neighbors for an instance and updating single weights for each instance using local neighborhood of k nearest neighbors. Procedure of weights adaptation is performed in several iterations I specified by a user. Each iterations repeats steps (2)-(4):

(1) Set equal pairwise weights, $\overline{w}_{r,s}^{f_j} = 1/N$, where N is the number of features;
(2) Compute distances using pairwise weights and find k nearest neighbors for each instance (Formula 60);
(3) Update single weights according to dispersion of feature values in local neighborhood (Formula 63);
(4) Compute pairwise weights by means of feature weights (Formula 64);

In our implementation, bidirectional data partitioning and subsequent integration of the component classifiers can be performed in several alternative ways depending on the following optional settings:

- perform clustering inside classes to obtain subgroups of instances (default: clustering disregarding class labels to obtain subgroups);
- perform subgroups merging according to feature weights profile (used as a default option, alternative: treat subgroups as final groups of instances to build component classifiers);
- perform feature selection by means of feature weights inside each group (used as a default option, alternative: skip feature selection by means of feature weights);

- perform feature selection using any of WEKA's feature selection technique inside each group (default: use WEKA's Correlation-based feature selection (CFS), alternative: use a different feature selection technique or skip feature selection);
- use group labels as new class labels to perform selection among component classifiers as an integration scheme (default: the component classifier is selected via finding the nearest group of instances to the test instance).

In addition, one can assess class separability and classification complexity on original data and inside each group. After partitioning, with or without feature selection, each group of instances can be recorded as a separate data file in WEKA's format for further analysis.

BDP implementation required modification of some basic WEKA's components in order to incorporate storage and manipulation for instance-feature pair associated weights called *single weights*. Therefore, distance metrics and clustering techniques used in WEKA had to be updated to use instances with feature weights.

For the experiments with BDP, only Manhattan distance function and two clustering techniques DBSCAN and *k*-Means underwent modifications related to handling single weights.

Any WEKA's feature selection technique can be chosen in order to build the component classifiers of BDP. In addition, component classifiers can be built using a local *feature weight profile* (an average or median feature weights of the correspondent data partition called *group* of instances). In this case, a threshold for feature weights is applied in order to perform feature selection. A threshold is an average or median cut-value calculated over a local feature weight profile.

Due to specifics of weight adaptation and subgroups merging procedure in BDP, high weights in a local feature weights profile can be assigned to features that are not individually predictive in this group, because they had small dispersion of values in different classes that fall into the highly intersected intervals of values in two classes. Therefore, local feature selection with WEKA's techniques can be optionally performed on top of somewhat redundant feature selection by feature weights.

For data sets with known globally irrelevant features, especially, if the number of features extremely large with respect to the number of instances, BDP's work can be facilitated by application of global feature selection. In this case, BDP can be used as a base classifier inside WEKA's meta-classifier with embedded feature selection (*AttributeSelectionClassifier*).

In our implementation, weights adaptation via local neighborhood is an optional step. For comparison purposes, grouping of instances can be performed without weights adaptation.

There are several numeric parameters in BDP: the number of nearest neighbors k (default: $= \sqrt{M}$, where M is a number of training instances), λ that controls strength of the incentive to distribute high weight among smaller number of features (default: $\lambda = 0.2$). The group sizes are controlled by

introduction of weight W_l to the criterion $Q(c)$ (Formula 59), where $\{W_l = M_l^2\}_1^L$ to encourage the nearly equal sized groups. The number of the retained features in a subset for in each group is controlled by λ.

We have used empirical values for initial parameter settings based on recommendations provided in related studies, whenever available. For example, argumentation for I, λ, and k settings is given in (Friedman & Meulmann, 2002). In the future, best parameter settings can be found with parameter tuning on a validation set or using a designated parameter tuning techniques available in WEKA, for example, *CVParameterSelection* technique.

The clustering algorithms in BDP are deterministic. Therefore, we have experimented with two clustering techniques that can use a weighted inverse exponential distance function: (1) DBSCAN, which does not require specification for the number of clusters, and (2) *k*-Means, which does.

WEKA's filter for removing "useless" variables that vary too much or too little is used in BDP implementation. A filter removes variables taking nearly constant values from consideration. We have used 95% as a value frequency threshold for SEER cancer data sets, and default 99% for other data sets. This filter also works for missing values if a missing value is treated a separate value. Nominal features that vary too much, for example, a patient's ID feature, do not possess any discriminative power to distinguish classes. We have used WEKA's filter with the threshold set to 99% in order to remove features that take a different nominal value in 99% of all cases. In addition, we have filtered out inconsistent instances that are identical instances in all features but class variable. We have implemented a new filter in WEKA called *RemoveInconsistent* among supervised instance filters in order to accomplish this task. Data set is filtered twice: in the beginning and before building local models. Features that vary too much or too little may appear locally as a result of partitioning and should be removed.

The basic scheme for BDP technique is presented in (Figure 11). Iterative weight adaptation if followed by weighted distance based clustering which obtains subgroups of interests (due to clustering methods specifics). These subgroups are merged to obtain final local groups, or given directly as an input to a meta-classifier. This BDP scheme can be realized in exactly 2^7=128 ways depending on combination of seven key procedures: (1) weight adaptation (WA) (alternative: equal weights), (2) clustering inside classes (CIC) (alternative: clustering disregarding class labels), (3) clustering with preset number of clusters (NumClust) (using *k*-Means) (alternative: preset radius parameter for density (using DBSCAN)), (4) clusters merging based on weights profile (IPA-merge) (alternative: treat clusters as final grouping of instances), (5) feature selection using feature weights profile (FSbyFW) (alternative: do not perform feature selection by feature weights profile), (6) local feature selection (LFS) (alternative: skip local feature selection), (7) integration using meta-classifier that maps clusters to classes (MetaC2C) (alternative: weighted inverse exponential distance-based integration based on nearest group (WIED-NN)).

Each procedure can be assigned a binary value: “yes” or “1” to perform the procedure in BDP and “no” or “0” to perform the alternative procedure. Among all 128 combinations we have selected a few most interesting ones to perform BDP, as shown in Table 7.

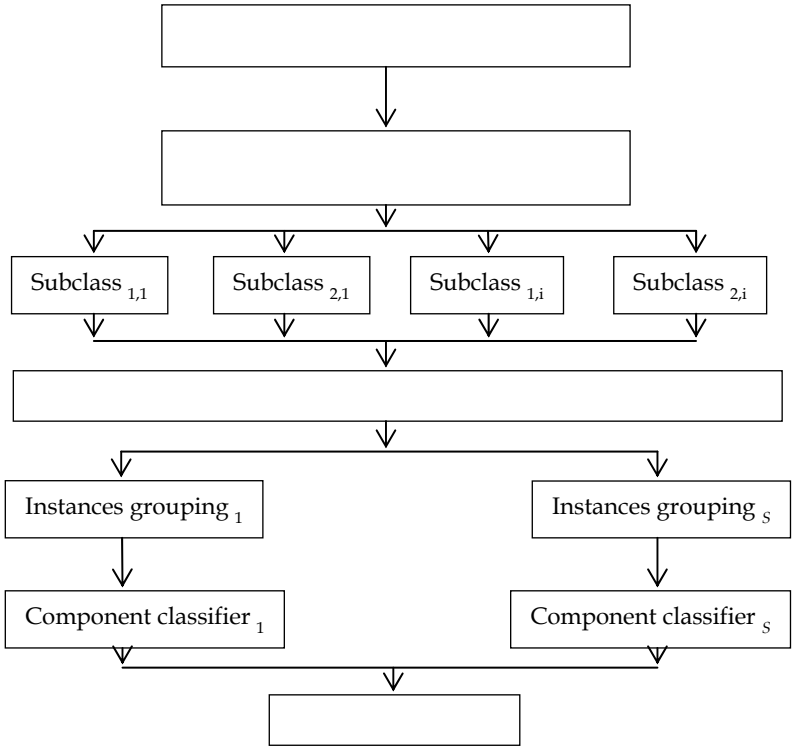

FIGURE 11 General Bidirectional Data Partitioning (BDP) scheme.

Enumerated BDP scheme realizations do not include a number of alternatives that can be introduced by selection of feature selection method for LFS, for example, CFS, ReliefF, InfoGain, or SU. Even more alternatives can be introduced by selection of a basic classification method, for example, J48, NaiveBayes, *k*-NN, SVM, etc. In addition, BDP itself can be used as a base classifier of any WEKA’s meta-classifier, for example, *AttributeSelectedClassifier*, *RandomSubSpace, Bagging,* or *AdaBoostM1*. A meta-classifier can also be used as a base classifier in BDP. In the experimental sections, the abbreviation specified in Table 7 will be supplied with additional details, including feature selection and base classifier names.

In this work we do not aim to study systematically all possible variants along with parameter tuning. For the experiments one or several schemes are chosen based on empirical observations.

BDP scheme has several parameters crucial for instances grouping: (1) subgroups are obtained based on the number of clusters if *k*-Means clustering is used, or the □-radius parameter if DBSCAN is used, and (2) subgroups merging into final groups depends on the IPA threshold, where IPA is an Importance Profile Angle measures over feature weights profiles in subgroups. Default

parameter for IPA threshold is 0.5 based on empirical results (Apte *et al.*, 1998). A more appropriate parameter value that regulates the number of final groups can be chosen after several runs. It has been done manually in our experiments. Default radius parameter in DBSCAN is set to an average distance to k^{th} neighbor across the data set, where $k = \sqrt{M}$.

TABLE 7 Bidirectional Data Partitioning (BDP) implementation schemes.

Scheme	WA	CIC	NumClust	IPA-merge	FSbyFW	LFS	MetaC2C
BDP-1	1	0	0	1	0	0	0
BDP-2	1	1	0	1	0	0	0
BDP-3	1	0	1	1	0	0	0
BDP-4	1	1	1	1	0	0	0
BDP-5	1	0	0	0	0	0	1
BDP-6	1	1	0	0	0	0	1
BDP-7	1	0	1	0	0	0	1
BDP-8	1	1	1	0	0	0	1
BDP-9	1	0	0	1	0	1	0
BDP-10	1	1	0	1	0	1	0
BDP-11	1	0	1	1	0	1	0
BDP-12	1	1	1	1	0	1	0
BDP-13	1	0	0	0	0	1	1
BDP-14	1	1	0	0	0	1	1
BDP-15	1	0	1	0	0	1	1
BDP-16	1	1	1	0	0	1	1
BDP-17	0	0	0	1	1	0	0
BDP-18	0	1	0	1	1	0	0
BDP-19	0	0	1	1	1	0	0
BDP-20	0	1	1	1	1	0	0
BDP-21	0	0	0	0	1	0	1
BDP-22	0	1	0	0	1	0	1
BDP-23	0	0	1	0	1	0	1
BDP-24	0	1	1	0	1	0	1
BDP-25	0	0	0	1	0	1	0
BDP-26	0	1	0	1	0	1	0
BDP-27	0	0	1	1	0	1	0
BDP-28	0	1	1	1	0	1	0
BDP-29	0	0	0	0	0	1	1
BDP-30	0	1	0	0	0	1	1
BDP-31	0	0	1	0	0	1	1
BDP-32	0	1	1	0	0	1	1

Integration of the component classifiers is implemented in two ways: distance-based integration and integration using a meta-classifier.

Distance-based integration includes the following steps:

- calculate distances,
- find k-neighborhood,
- consider to which group neighbors belong,
- apply a group classifier where majority of k-NN belong,
- apply the respective filter.

Meta-classifier is built on group labels and assigns instance to the corresponding class thereafter.

In BDP, distance is used during weights adaptation, pre-estimation for DBSCAN, DBSCAN or *k*-Means, and integration of classifiers.

5.4 Chapter summary

This chapter describes the Bidirectional Data Partitioning technique that searches for local homogeneous regions in feature subspaces with simplified discrimination between classes, builds classifiers, one per local region, and combines them in an ensemble for prediction. The method is derived from the subspace clustering technique COSA and uses a discrimination criterion based on a class separability measure. The theoretical background provided includes formulation of an optimization task and choice of weight functions.

Bidirectional Data Partitioning is based on the bottom-up search. It divides a data set into local regions to build local models simultaneously finding the feature subspaces such that subgroups of instances from the same class have smaller intra class distances, and at the same increasing interclass distances. The goal is to uncover the local structure in subspaces, where discrimination between classes will be simplified. Therefore subgroups are merged thereafter for better class separability. The size of local regions is controlled, so that they would not be too small for building local predictive models. The overlapped regions are joined. The local regions found during the learning phase are described as convex hulls in local subspaces and this description is used during the classification phase. A new unclassified instance falls into one or more regions, thus an appropriate local model to classify this instance is selected. k-NN based procedure is used to select an appropriate group and associated classifier. Alternatively, subgroups are used as new classes in meta-classifier. This meta-classifier assigns a new instance a label, and then determines which local classifier to use for this instance. In Fradkin (2006) a simplified version of this meta-classifier is used when clustering is performed within classes, local models are not constructed, only translation of labels is performed.

6 EMPIRICAL EVALUATION OF BIDIRECTIONAL DATA PARTITIONING

This chapter describes an empirical evaluation of the suggested decomposition strategies at all stages. Characteristics of selected data complexity measures are studied on synthetic and benchmark data sets in order to evaluate their descriptive abilities regarding data structure, and in particular, unstable feature relevance. Decompositions by means of class encoding and decompositions guided by a class separability criterion are studied. A detailed study on synthetic and benchmark data sets is performed for Bidirectional Partitioning. This includes evaluation of the weights adaptation scheme, decomposition into subgroups of instances representing subclasses, joining subgroups with similar weights profiles, building local classifiers, and a classifiers selection scheme. Related data pre-processing and experimental setup topics are briefly covered in the beginning.

6.1 Evaluation of class separability measures

Unstable feature relevance characterizes a general case of classification heterogeneity. According to its description given in Subsection 3.2.1, data instances can be grouped according to feature relevance profile. In other words, two groups may have different subsets of relevant features; hence, locally irrelevant features can be ignored. Local regions in data presented by groups of instances in the subset of relevant dimensions assumed to have better class separability and decreased classification complexity. Class separability

measures, therefore, can be used to produce a criterion function in search of those local regions, especially, considering the fact, that majority of feature selection techniques used in classification tasks are based on class separability criteria and related measures.

Before modeling feature space heterogeneity, it is useful to evaluate to which extent various class separability and other complexity measures respond to elimination of irrelevant features as well as their ability to characterize geometrical structure of data. In order to give a geometrical interpretation to the abstract notion of data structure, a collection of synthetic and benchmark data sets with known properties is created. Focusing on data characteristics such as class shapes, class boundaries, margins between classes, density and disconnected regions, contributes to understanding bidirectional data partitioning performance and in case of unstable (local) feature relevance and class heterogeneity. Studying these cases with class separability and other data complexity measurements helps to interpret these measurements on data with unknown properties. Some of the complexity measures may indirectly give an indication of heterogeneity presence.

6.1.1 Synthetic and benchmark data sets

We would like to investigate classification problems of different geometrical complexity with class separability and other complexity measures.

Several data sets with known irrelevant features are described below. Those include synthetic data sets with irrelevant features and benchmark data sets with irrelevant features added for this study. All irrelevant features are artificially created according to definitions of feature relevance given in Subsection 2.1.3.

Data properties under study include: (a) linearity and nonlinearity of boundaries between classes, (b) even or uneven density within class, (c) equal or unequal density in different classes, (d) wide or narrow margins between classes, (e) Gaussian or non-Gaussian subclasses. Each data set being analyzed is representative to several properties of interest.

This study is limited to data sets with the same type of features (continuous numeric), though in reality matters are complicated by a mixture of data types. Unbalanced class distributions and missing values are also avoided.

Below the data sets used in this study are described. Graphical presentation of all synthetic data sets is provided in Appendix 2.

GaussS-2 is a synthetic data set with two Gaussian classes, 2500 instances in each, almost completely separable, with means $\mu_A = 5$, $\text{stdev}_A = 1$ denoted as G(5;1); $\mu_B = 10$, $\text{stdev}_B = 1$ denoted as G(10;1). **GaussS-2+1U** has one additional unimodal irrelevant feature that has a Gaussian distribution G(7.5;3). **GaussS-2+1B** has one additional bimodal irrelevant feature that is a mixture of G(6;3) and G(9;3). **GaussS-2+1M** has one irrelevant feature that is a mixture of G(2;3), G(5;3), and G(8;3). **GaussS-2+1** has one irrelevant feature uniformly distributed in the interval [0…15] denoted as U(0;15). **GaussS-2+all** includes all of the above irrelevant features.

Gauss-8 (Blayo *et al*., 1995) is a data set with 8 continuous features generated according to Gaussian distributions in two classes. There is a set of seven databases corresponding to the same problem with dimensionality ranging from 2 to 8. The class 0 is represented by a multivariate normal distribution G(0.0, 1.0) in all dimensions, and the class 1 is given by G(0.0, 2.0) in all dimensions. There are 5000 instances, 2500 in each class. The center of gravity is the same for two classes, which makes them heavily overlapped. The theoretical error is 9%. **Gauss-8+10** is a modified version of Gauss-8 with 10 irrelevant features, U(-5; 5). It has heavily interleaved classes, approximately equal covariances in classes, equal density in both classes.

FourSubclass-2 data set has two Gaussian subclasses per each of two classes. The first class cl0 is composed of two Gaussian distributions: subclass 1 as G(5.0, 1.0) in both features f_1 and f_2, and subclass 2 as G(10.0, 1.0) in both features. Two subclasses of class cl1 are f_1 - G(10.0, 1.0), f_2 - G(5.0, 1.0) and f_1 - G(5.0, 1.0), f_2 - G(10.0, 1.0). Each subclass is represented by 250 instances in both training and test sets. This data set has a Bhattacharyya upper bound on minimum Bayes error 0.2415. In **FourSubclass-2+5G** there are 5 irrelevant features created with G(7.5, 2.0). In **FourSubclass-2+5U** there are 5 irrelevant features created with U(2.0, 13.0). **FourSubclass-2+10** includes both types of irrelevant features, 5 of each.

Clouds-2 data set (Blayo *et al*., 1995) has 2 classes, one of which has three Gaussian subclasses. One of the subclasses is heavily interleaved with the other Gaussian class, while the other two are partially interleaved. The original data set has 5000 instances, 2500 in each of two classes, 2 continuous numeric features. The theoretical error is 9.66%. **Clouds-2+10** data set has 10 irrelevant features added, U(-3; 3).

Concentric-2 data set (Blayo *et al*., 1995) consists of two classes: one uniformly distributed within a concentric area, and another class surrounds it without overlapping. This is an example of narrow margin between classes, nonlinear boundary, equal and even density in classes. The original data set has 5000 instances, equal-size classes, and 2 features. The theoretical error is 0%. **Concentric-2+10** has additional 10 irrelevant features, U(-5; 5).

Spirals-2+5 data set is a modified version of 2-dimensional data presented in (Lindenbaum *et al*., 1999). It has 3 irrelevant features, uniformly distributed, U(0; 12). In the original **Spirals-2** data set two features take values in the interval (0…20). This data set is an example of nonlinear class boundaries with narrow margins.

Fourclass-2+7M is a data set that presents a case, where irrelevant features obtained from mixed distributions partially capable do discriminate classes, and two relevant features show nonlinear boundary between classes with a wide margin. There are 4 classes (271, 266, 274, and 327 instances), 9 numeric features, 2 relevant and 7 irrelevant features. **Fourclass-2** is a data set with two relevant features retained. This data set is a class-balanced 12% sample of the original data set used in (Bernadó-Mansilla & Ho, 2005), which has an equal

density unbalanced classes. In Fourclass-2 classes have unequal density. **Fourclass-2+3** has 3 uniformly distributed irrelevant features, U(0;1).

Birch-2+8 data set is generated using BIRCH data generator in WEKA ("grid" pattern is used). The data set has 3 classes and 10 continuous features, only two of which are partially discriminative. Features #1 and #2 each discriminate a different class from the other two and fully discriminative together; in other features classes are heavily interleaved. There are 11055 instances, classes are balanced: 3695, 3782, 3578 instances respectively. **Birch-2** is the same data set with only two relevant features retained.

RBF-10+10 data set is generated in WEKA using RandomRBF generator. The data set is created by first creating a random set of centers for each class following the number of specified centroids. Each center is randomly assigned a weight, a central point per feature, and a standard deviation. To generate new instances, a center is chosen at random taking the weights of each center into consideration. Feature values are randomly generated and offset from the center, where the overall vector has been scaled so that its length equals a value sampled randomly from the Gaussian distribution of the center. The particular center chosen determines the class of the instance. RandomRBF generated data contains only numeric features. In RBF1-10 there are 5000 instances, 2 classes, 2327 and 2683 instances accordingly, 10 features distributed in the interval (-2...2.5), the number of centroids is 50. In addition, there are 10 irrelevant uniformly distributed features, U(-5; 5). **RBF-10** data set is the same except for these 10 irrelevant features. This data set is an example of Gaussian subclasses in heavily interleaved classes.

RDG-10+10 data set have been obtained using WEKA's random data generator RDG1. It creates data randomly by producing a decision list consisting of rules. Instances are generated randomly one by one. If the decision list fails to classify the current instance, a new rule according to this current instance is generated and added to the decision list. RDG-10+10 has 10 continuous relevant features distributed in the interval (-1.5…2.5) and additional 10 irrelevant features, U(-5; 5). The maximum and minimum numbers of tests in rules are set to 10 and 1 accordingly. There are 2 classes, 2548 and 2452 instances. **RDG-10** data set is the same except for the 10 irrelevant features. Data more elongated in irrelevant dimensions compared to relevant dimensions, but there is no visual distinction between relevant and irrelevant features in terms of class boundaries. Classes are heavily interleaved.

6.1.2 Experimental settings

For multi-class problems, pairwise decomposition has been performed and the results are averaged. Probability estimates to calculate class separability measures are implemented in connection to previous usage in data mining and pattern recognition research papers. Sample mean and covariance estimates are used.

Data sets used in this study include only continuous features due to various difficulties associated with computation of parametric measures

originated from multivariate statistical methods on data with mixed feature types (Kowalsky, 1972). All continuous features have been normalized. Continuous numeric features have been discretized for calculation of information-theoretic measures using unsupervised discretization with a number of bins equal to $\frac{1}{2}\sqrt{M}$, where M is a training set size. Supervised discretization (Fayyad & Irani, 1993) detects uniformly distributed irrelevant features and assigns them a constant value, therefore unsupervised discretization has been chosen.

Distance computation for continuous features in complexity measurements, as well as in intra/inter class distance ratio, is based on a non-normalized Euclidean distance function (Orriols-Puig *et al*., 2010).

For this study, all class separability measures have been implemented in WEKA (Witten & Frank, 2005) and MATLAB. All related experiments have been also performed in WEKA, except for complexity measures that were calculated using Data Complexity Library DCoL v1.1 (Orriols-Puig *et al*., 2010). Synthetic data have been generated using WEKA and MATLAB.

Classification error rate is reported over a single run of 10-folds cross-validation. Unless specified, WEKA and DCoL techniques have been used with default parameters. Euclidean distance is used in k-Nearest Neighbor classifier, k = 10. Normalized data sets versions are used for classification and feature ranking.

6.1.3 Class separability, complexity, and irrelevant features

In Section 4.3 different class separability and other complexity measures have been discussed. Considering applicability restrictions of various measures, a set of measures is selected to evaluate their sensitivity to irrelevant features and their descriptive abilities on different geometrical data structures. Class separability measures are candidates for a criterion function in bidirectional data partitioning. Complexity measures not based on class separability provide additional insights about data and contribute toward results interpretation.

Eight measures based on class separability (1-8) and eight (9-16) based on other criteria are chosen:

1. Normal Information Radius ($NIR_{A,B}$);
2. Bhattacharyya distance ($a_{A,B}$);
3. Kullback-Leibler distance ($KL_{A,B}$);
4. divergence ($DIV_{A,B}$);
5. scatter matrices based criterion (J_1);
6. scatter matrices based criterion (J_2);
7. Fisher's discriminant ratio ($F_{A,B}$);
8. the ratio of average intra/inter class nearest neighbor distance ($\partial_{A,B}$);
9. fraction of points on the class boundary (N1);
10. the leave-one-out error rate of the one-nearest neighbor classifier (N2);
11. the nonlinearity of the one-nearest neighbor classifier (N3);
12. the fraction of maximum covering spheres (T1);

13. the average number of points per dimension (T2);
14. the minimized sum of the error distance of a linear classifier (Linear SMO SVM) (L1);
15. the training error of a linear classifier (Linear SMO SVM) (L2);
16. the nonlinearity of a linear classifier (Linear SMO SVM) (L3).

In original formulae some measures are directly proportional to class separability (J_2 and $\partial_{A,B}$), while others are inversely proportional to class separability ($NIR_{A,B}$, $a_{A,B}$, $KL_{A,B}$, $DIV_{A,B}$, J_1, and $F_{A,B}$). Therefore, we have transformed the latter so that all measures are inversely proportional to the class separability, and criterion based on these measures would have to be minimized.

Class separability measures cannot be compared directly due to different measurement scales. Most of the measures are unbounded. Therefore, we observe the relative change after irrelevant features elimination and provide interpretation of the results for each particular data set. Most measures are not invariant to the number of features in the data set and need to be adjusted accordingly to observe a change in class separability. Absolute values of all measures depend on the number of features used. $KL_{A,B}$, $DIV_{A,B}$, and $F_{A,B}$ can be computed for an individual feature in order to find a maximum discriminative one (the rest of features are just ignored), but in this study $KL_{A,B}$, $DIV_{A,B}$ and $F_{A,B}$ are computed over all features.

In order to see how the number of features affects the measures, Gauss-8 data set is used with different number of features, from 2 to 8. This data set originally has been created in order to study classifiers behavior for different dimensionalities of the feature space, for heavily overlapped distributions and for nonlinear separability. All features in this data set follow the same distribution. The results are presented in Table 8. In addition to class separability and complexity measures used in further experiments, results on intra class and inter class distances as well as their ratio and difference (intra class – β * inter class, β = 2) are shown. Measures for $\partial_{A,B}$ and distances are provided for Euclidean distance (Eucl) and for Inverse Exponential distance with equal weights (InvExp).

The results have shown that intra class distance increases with increased number of dimensions, while the inter class distance decreases encouraging low-dimensional solutions in BDP. The difference between intra and inter class distances increases. The ratio of these distances increases with the number of dimensions increased. Despite the averaging we can still observe that the measures are still dependent on the number of dimensions used, hence, the improvement obtained after elimination of irrelevant features should be partially credited to this effect. Most of the measures are not a sum of individual feature's items, therefore averaging would not derive a measure in one feature (knowing that all features in this data set are synthetically created in the same way).

Being scaled down to average per feature, some measures show slight decrease in class separability with the number of dimensions increased due to

the *curse of dimensionality* problem. Most techniques dealing with metric space cannot resist this problem: in high dimensions all points in the metric space become nearly equidistant (Parsons *et al*., 2004). On the other hand, research on high-dimensional data projections into low dimensions states that under certain conditions of a weak regularity for a high-dimensional distribution, its low-dimensional linear projections can turn into a mixture of centered spherically symmetrical Gaussian distributions (Dümbgen & Zerial, 2011; Jimenez & Landgrebe, 1999) that can affect measured class separability.

As shown in Table 8, being scaled down to average per feature, some measures have a tendency to change slightly with the number of dimensions increased. This effect can be partially credited to aforementioned phenomena.

Both $NIR_{A,B}$ and $a_{A,B}$, generalizations of Mahalanobis distance, have a tendency to decrease with a number of dimensions, either adjusted by the number of features or not, which means that class separability provided by $NIR_{A,B}$ and $a_{A,B}$ increases with the number of dimensions. The original measures in Formulae (34) and (35) correctly identify that classes are completely overlapped with values 0.3156 and 0.2170 accordingly, but those values misleadingly increase with the number of dimensions. These measures can possibly lead to biased results in high-dimensional problems.

$KL_{A,B}$ and $DIV_{A,B}$ are related measures. In particular, symmetric Kullback-Leibler distance is twice of divergence. As measures are unbounded, their values considerably decrease with the number of dimensions in both variants, adjusted by features number or not. The original measures, Formulae (36) and (37), take values close to 0, which stands for completely overlapped classes. The less dimensions used in Gauss-8 data, the closer to 0 the values get. Hence, some bias in high dimensions is present. In original measures, though, it appears on a smaller scale.

J_1 and J_2, scatter matrices based measures, $F_{A,B}$, Fisher's linear discriminant, and $\partial_{A,B}$, intra- inter class distances ratio, have also demonstrated a bias resulted from the properties of high-dimensional data projections. J_1, J_2, and $F_{A,B}$ slightly increased in higher dimensions showing better class separability for the problem. $\partial_{A,B}$ shows a tendency to decrease.

To conclude, in Gauss-8 data set measures $NIR_{A,B}$, $a_{A,B}$, $KL_{A,B}$, $DIV_{A,B}$, J_1, $F_{A,B}$, and $F_{A,B}$ have shown more class separability in higher dimensions, while $\partial_{A,B}$ (measured as equally weighted inverse exponential distance) has shown less class separability, due to the λ factor.

Complexity measures N1, N2, N3, T1, L1, L2, L3 all take values in the interval (0..1), with 0 corresponding to better class separability, except for T2, which is a ratio of instances and features. N1, fraction of points on the class boundary, N2, leave-one-out error rate of the 1-NN classifier, and N3, the nonlinearity of the 1-NN classifier, are decreasing in higher dimensions showing more class separability. T1, the proportion of retained adherence subsets, which involves distance computation, remains nearly unchanged. T2, the average number of points per dimension, decreases naturally in higher dimensions. Among measures derived from error rates of a linear classifier, L1

decreases very slightly, L2 and L3 practically remain unchanged.

TABLE 8 Class separability and complexity measures in different dimensionality. The smaller the value, the better class separability. Complexity measures values are all spanned in the interval (0,1).

Measures	Number of features retained in Gauss-8 data set						
	2	3	4	5	6	7	8
	Not scaled by the number of features						
$NIR_{A,B}$	3.1682	0.7962	0.2267	0.1280	0.0889	0.0673	0.0551
$a_{A,B}$	4.6082	3.0383	2.2554	1.7723	1.4888	1.2684	1.1228
$KL_{A,B}$	6640.10	4291.07	3171.87	2521.11	2132.26	1822.42	1606.93
$DIV_{A,B}$	3320.05	1430.36	792.968	504.222	355.377	911.211	803.467
J_1	$1.8*10^4$	$0.3*10^4$	$0.2*10^4$	$0.2*10^4$	$0.2*10^4$	$0.1*10^4$	$0.1*10^4$
J_2	$3.8*10^4$	$1.0*10^4$	$1.0*10^4$	$1.1*10^4$	$1.3*10^4$	$1.0*10^4$	$1.0*10^4$
$F_{A,B}$	155.578	19.8597	16.6683	15.6895	15.2234	10.8071	9.6500
$\partial_{A,B}$, InvExp	0.0038	0.0116	0.0201	0.0279	0.0344	0.0405	0.0456
$\partial_{A,B}$, Eucl	0.3825	0.5846	0.6620	0.7173	0.7517	0.7794	0.7944
	Scaled by the number of features						
$NIR_{A,B}$	1.5841	0.2654	0.0567	0.0256	0.0148	0.0096	0.0069
$a_{A,B}$	2.3041	1.0128	0.5638	0.3545	0.2481	0.1812	0.1403
$KL_{A,B}$	3320.05	2145.53	1585.94	1260.55	1066.13	260.346	200.867
$DIV_{A,B}$	1660.02	715.178	396.484	252.111	177.689	130.173	100.433
J_1	$0.9*10^4$	$0.09*10^4$	$0.05*10^4$	$0.04*10^4$	$0.03*10^4$	$0.02*10^4$	$0.01*10^4$
J_2	$1.9*10^4$	$0.3*10^4$	$0.2*10^4$	$0.2*10^4$	$0.2*10^4$	$0.1*10^4$	$0.1*10^4$
$F_{A,B}$	77.789	6.6199	4.1671	3.1379	2.5372	1.5439	1.2063
$\partial_{A,B}$, InvExp	0.0019	0.0039	0.0050	0.0056	0.0057	0.0058	0.0057
$\partial_{A,B}$, Eucl	0.1912	0.1949	0.1655	0.1435	0.1253	0.1113	0.0993
	Not scaled by the number of features						
N1	0.5213	0.4533	0.3979	0.3561	0.3303	0.3147	0.3147
N2	0.3499	0.3089	0.2702	0.2406	0.2098	0.1944	0.1836
N3	0.3797	0.3309	0.2828	0.2603	0.2446	0.2466	0.2463
T1	0.9730	0.9952	0.9998	1.0000	0.9996	0.9994	0.9996
T2	2499.5	1666.3	1249.7	999.80	833.17	714.14	624.87
L1	0.9997	0.9990	0.9986	0.9985	0.9985	0.9977	0.9973
L2	0.4999	0.4163	0.4999	0.4999	0.4999	0.4999	0.4999
L3	0.5000	0.4315	0.5000	0.5000	0.5000	0.5000	0.5000
Intra dist, InvExp	0.0034	0.0095	0.0162	0.0218	0.0262	0.0303	0.0340
Inter dist, InvExp	0.8768	0.8224	0.8042	0.7817	0.7600	0.7485	0.7456
Ratio	0.0038	0.0116	0.0201	0.0279	0.0344	0.0405	0.0456
Diff, β=2	-1.7502	-1.6353	-1.5922	-1.5416	-1.4938	-1.4667	-1.4571

Complexity measures, except for T2, are bounded, and their change appears on a different scale. Some of them show slight decrease in higher dimensions that is a better class separability.

In higher dimensions, theoretical error rate for this data set decreases (Blayo *et al.*, 1995). We can expect that class separability has a tendency to increase. $KL_{A,B}$, $DIV_{A,B}$, and $\partial_{A,B}$ measures came up with an opposite tendency. This impact of high dimensions should be taken into account as we observe the change in class separability after elimination of irrelevant features.

In further computations, class separability measures are not scaled down by the number of features over the number of features, as this step does not stabilize the results. Also the original values of $NIR_{A,B}$, $a_{A,B}$, $KL_{A,B}$, $DIV_{A,B}$, J_1, and $F_{A,B}$ are used, which are inversely proportional to class separability. These measures, except for J_1, are bounded by a 0 value for a complete overlap, which is convenient for results interpretation. $\partial_{A,B}$ takes values in the interval (0..1), approaching 0 in case of better class separability, approaching 1 in case of possibly interleaved classes. J_1 and J_2 are unbounded measures, therefore one can only observe a relative change.

Next, selected class separability measures are evaluated on Gaussian data without irrelevant features. All data sets in this experiment are two-dimensional, have two classes, 1000 instances each. **Gauss-2-sep** is a data set with distributions in class 1 G(5.0, 1.0) – feature 1, and G(10.0, 1.0) – feature 2, and in class 2 G(10.0, 1.0) – feature 1, and G(2.0, 1.0) – feature 2. **Gauss-2-sep** is an example of linear separability with wide margins between classes. **Gauss-2-one** is a data set with distributions in class 1 G(5.0, 1.0) – feature 1, and G(10.0, 1.0) – feature 2, and in class 2 G(10.0, 1.0) – both feature 1 and 2. **Gauss-2-one** is linearly separable in feature 1, there is a minor intersection between classes, a narrow margin. **Gauss-2-onesep** is a data set with distributions in class 1 G(5.0, 0.5) – feature 1, and G(10.0, 0.5) – feature 2, and in class 2 G(10.0, 0.5) – both feature 1 and 2. **Gauss-2-onesep** is linearly separable in feature 1 with no intersection between classes. **Gauss-2-ov** has G(10.0, 0.5) in both dimensions, both classes, that means classes are completely overlapped. Table 9 holds the results.

TABLE 9 Class separability and margins between classes.

Data	Class separability measures							
	$NIR_{A,B}$	$a_{A,B}$	$KL_{A,B}$	$DIV_{A,B}$	J_1	J_2	$F_{A,B}$	$\partial_{A,B}$, InvExp
Gauss-2-sep	5.3303	6.5491	0.0046	0.0093	13.0974	0.1607	3232.61	0.0030
Gauss-2-one	4.0246	3.2028	0.0024	0.0047	6.4056	0.7028	1847.56	0.0041
Gauss-2-onesep	5.0959	12.3223	0.0026	0.0051	24.6433	0.2840	14004.7	0.0031
Gauss-2-ov	0.0394	0.0013	0.0000	0.0000	0.0013	1616.15	0.1180	0.0063

Values of $NIR_{A,B}$, $a_{A,B}$ are not close to 0 for Gauss-2-sep, the case of complete separability with wide margins, which is right. Values for $KL_{A,B}$ and $DIV_{A,B}$ are highest of all cases, that is a non-zero class separability. Values of J_1 and $F_{A,B}$ are not nearly close to 0, which corresponds to non-zero class separability. J_2 and $\partial_{A,B}$ are expected to have small values in case of complete separability, and they do show small values in this case. In case of linear

separability and wide margins between classes, Gauss-2-sep, all measures were capable to detect it.

In case of one discriminative features and narrow margin between classes, Gauss-2-one, $NIR_{A,B}$, $a_{A,B}$, J_1, and $F_{A,B}$ produce smaller value for class separability compared to Gauss-2-sep, but this value is far from 0. $KL_{A,B}$ and $DIV_{A,B}$ return smaller value than in case of Gauss-2-sep and Gauss-2-onesep, but higher than in Gauss-2-ov. J_2 and $\partial_{A,B}$ produced proper values, but smaller compared to separability with wide margin between classes.

In case of one discriminative features and wide margin between classes, Gauss-2-onesep, $a_{A,B}$, J_1, and $F_{A,B}$ have responded with a highest value. Possibly, feature relevance distributed among fewer features is more favorable for these measures. $NIR_{A,B}$ indicates class separability with about the same value as in Gauss-2-sep. $KL_{A,B}$ and $DIV_{A,B}$ provide unexpected results: wide margin between classes seems to be worth than narrow margin between classes in case of one separating feature.

In case of fully overlapped classes, Gauss-2-ov, where means coincide and variances are equal, all measures, except for $\partial_{A,B}$, indicate low class separability. Only $\partial_{A,B}$ can detect non-zero separability in presence of nonlinear decision boundaries. $KL_{A,B}$ and $DIV_{A,B}$ have value larger that in case of linearly separable classes in Gauss-2-sep.

To conclude, $\partial_{A,B}$, the intra- and inter-class ratio, respond with the most adequate results, which is easy to interpret.

Before proceeding with experiments on data sets of different geometrical complexity, we briefly outline specifics of class separability measures used in addition to the basic information provided in Section 4.3.

$NIR_{A,B}$, Normal Information Radius, is defined in the interval [0;∞) and depends on the distribution of feature values. The value of $NIR_{A,B}$ is different before and after normalization. The larger $NIR_{A,B}$ is the better class separability. $a_{A,B}$, Bhattacharyya distance, is defined in the interval [0; ∞) measuring to which extent two classes overlap. Same as for $NIR_{A,B}$, $a_{A,B}$ depends on both means and covariances of two classes. The larger $a_{A,B}$, the better class separability. In case of equal covariances $a_{A,B}$ reduces to Chernoff bound on minimum Bayes error, which is $1/8$ of Mahalanobis distance. In cases of non-Gaussian classes, empirically, $a_{A,B}$ is still an informative class separability measure. $KL_{A,B}$ and $DIV_{A,B}$ take values in the interval [0;∞), where 0 corresponds to completely overlapped classes.

$KL_{A,B}$, Kullback-Leibler distance, and $DIV_{A,B}$, divergence, not only depend on two means of the pair of classes, but also on their covariances. For example, if two classes have coinciding means but different covariances, the divergence still can be far greater than 0. This makes sense, as class discrimination is possible to a certain extent due to the difference in variances. Direct relation of Bayes error and divergence is only possible for normally distributed classes with equal variances and prior probabilities, when divergence take form of Mahalanobis distance. It is also known that small variations of the differences between class means produces large changes in divergence and sometimes may

bring misleading results. Higher values of $KL_{A,B}$ and $DIV_{A,B}$ indicate better class separability taking values in the interval [0;∞).

J_1 and J_2 are unbounded measures. Class separability measured by J_1 and J_2 depends on both means difference and variances difference. The larger the value of J_1 the larger the inter class scatter, the better class separability. The smaller the value of J_2 the smaller the intra class scatter as compared to the inter class scatter, hence the better class separability. These measures are based on the separation of means, while the rest of class distribution information is ignored.

$\partial_{A,B}$ is a ratio on intra and inter class distances, where intra and inter class distances are computed as an average distance to the first nearest neighbor of the same class and different class distance correspondingly. In these experiments Euclidean distance normalized by the range of values in each feature is used. From Table 5.1 one can see that non-averaged variant of $\partial_{A,B}$ shows increased class separability in higher dimensions, just like other measures, and it will be easier to interpret the results using this measure. This measure take values in the interval (0..1). Low value of $\partial_{A,B}$ indicates that instances of the same class lie close to each other compared to any other class, and high value indicates that classes might be intersecting. In problems, where most instances appear next to class boundaries with narrow margins between them, this measure could be misleading. In multi-class problems this measure is averaged over pairwise decompositions. This means, for example, if two of three classes are intersecting while the third one is well-separated, the effect is averaged.

Results on class separability measures are provided in Table 10. Results, where class separability is improved as expected after elimination of irrelevant features are shown in bold.

For GaussS-2-all data, which is a complete linear separability case in all variants, with wide boundaries in GaussS-2, all measures expressed a similar behavior to what was demonstrated in Table 8. Additional dimensions in some cases contributed to class separability, most visibly in $NIR_{A,B}$. Additional non-discriminative features had no impact on separability as demonstrated by $a_{A,B}$, $KL_{A,B}$, $DIV_{A,B}$, J_1, and $F_{A,B}$. Only J_2 and $\partial_{A,B}$ have been able to detect irrelevant features properly. However, $KL_{A,B}$, $DIV_{A,B}$, and $F_{A,B}$ to the less extent, returned high individual feature merits for relevant features and small for irrelevant features. Therefore, these measures are more effective when applied for individual features in case the feature independence assumption is valid. Binomial irrelevant feature case was the most confusing for $F_{A,B}$ that tries to establish class mean.

Foursublass-2 is a non-linear class boundaries case with narrow margins. Additional irrelevant dimensions, uniform and Gaussian contribute to class separability according to all measures, except for $\partial_{A,B}$. $\partial_{A,B}$ is the only measure that can show benefits of eliminating irrelevant features. Presence of subclasses means substantial deviation from multimodal distribution, therefore measures based on evaluation of class means (centroids) fail.

Clouds-2 has partially overlapped Gaussian classes; one of classes is composed of three Gaussian distributions. The measure based on variances, J_2, has shown the most noticeable improvement, because uniformly distributed irrelevant features have the range approximately equal to the range of relevant features. $\partial_{A,B}$ has also shown increase in class separability. However, all other measures were not able to see the difference or went counterproductive ($NIR_{A,B}$).

TABLE 10 Class separability measures.

Data	$NIR_{A,B}$	$a_{A,B}$	$KL_{A,B}$	$DIV_{A,B}$	J_1	J_2	$F_{A,B}$	$\partial_{A,B}$
GaussS-2-all	15.8519	6.4777	0.0023	0.0045	12.9524	**1.6091**	3685.64	**0.2385**
GaussS-2+1	7.2149	6.2617	0.0021	0.0041	12.5228	**1.1052**	3630.12	**0.0602**
GaussS-2+1M	8.3181	6.3769	0.0022	0.0043	12.7530	**0.3565**	3655.47	**0.0602**
GaussS-2+1B	8.4044	6.3650	0.0022	0.0043	12.7292	**0.3340**	1220.07	**0.0525**
GaussS-2+1U	8.1545	6.2622	0.0021	0.0041	12.5240	**0.4163**	3630.18	**0.0548**
GaussS-2	5.4132	6.2614	0.0021	0.0041	12.5228	**0.1593**	3630.12	**0.0130**
Gauss-8+10	38.5893	0.9058	0.0006	0.0013	0.0050	2629.0	0.2038	**0.9729**
Gauss-8	18.1565	0.8907	0.0006	0.0012	0.0008	9728.7	0.1036	**0.7944**
Foursubclass-2+10	21.4552	0.7323	0.0003	0.0006	0.0096	1390.86	0.5756	**0.8268**
Foursubclass-2+5G	11.5477	0.7033	0.0002	0.0003	0.0055	1746.43	0.4783	**0.5706**
Foursubclass-2+5U	7.6588	0.6966	0.0002	0.0003	0.0040	1590.70	0.0974	**0.6816**
Foursubclass-2	0.2911	0.6816	0.0000	0.0001	0.0000	1264177.6	0.0001	**0.0740**
Clouds-2+10	20.285	0.2518	0.0005	0.0010	0.4875	**96.7449**	56.5910	**0.7601**
Clouds-2	2.4372	0.2430	0.0005	0.0009	0.4853	**4.0805**	56.5524	**0.1463**
Concentric-2+10	16.6454	0.2277	0.0009	0.0018	0.0040	2793.52	0.1055	**0.9279**
Concentric-2	0.3025	0.2100	0.0008	0.0017	0.0000	43350.02	0.0016	**0.0798**
Spirals-2+3	6.0510	0.1127	0.0002	0.0004	0.2231	**26.9939**	6.5963	**0.8302**
Spirals-2	0.9393	0.1118	0.0002	0.0004	0.2231	**10.0411**	6.5932	**0.1313**
Fourclass-2+3	7.7605	1.3123	0.0058	0.0116	1.7672	**12.6692**	88.0288	**0.5017**
Fourclass-2+7M	29.6570	12.4522	0.0286	0.0573	17.4358	1.7272	722.0508	**0.2182**
Fourclass-2	2.4322	1.3015	0.0056	0.0111	1.7620	**3.8305**	87.9588	**0.0532**
Birch-2+8	38.3900	15.0656	0.0024	0.0048	23.6531	**1.1779**	7649.02	**0.2743**
Birch-2	7.1526	12.2908	0.0011	0.0023	23.6077	**0.1213**	7648.93	**0.0073**
RBF-10+10	44.4200	0.4435	0.0004	0.0007	0.0891	**670.7541**	11.7520	**0.9657**
RBF-10	26.359	0.4263	0.0003	0.0007	0.0874	**106.5059**	11.7052	**0.4393**
RDG-10+10	33.6445	0.3805	0.0005	0.0011	0.5676	**54.2270**	10.1653	**0.9460**
RDG-10	15.7455	0.3652	0.0005	0.0010	0.5676	**26.7136**	10.1208	**0.8350**

Concentric-2, Spirals-2, and Fourclass-2 have even density, balanced classes, nonlinear boundaries, and non-Gaussian classes. In this case, most measures evaluating means and covariances /variances of two classes would give unreliable results.

Concentric-2+10 appeared to be one of the hardest cases of all used, because any of the measures were able to detect improvement in class separability except for $\partial_{A,B}$. In Concentric-2 the boundaries are nonlinear and the margin is narrow (touching). The distribution is uniform inside the concentric areas. Additional features were considered as somewhat informative, hence, class separability is better in presence of irrelevant features.

Spirals+2 has nonlinear boundaries with narrow margins. Most measures expressed a similar behavior as in Concentric-2+10. However, J_2 and $\partial_{A,B}$ were able to see increase in class separability after elimination of irrelevant features. $NIR_{A,B}$ shows more class separability in presence of irrelevant features, the other measures practically remained unchanged.

Fourclass+2 has nonlinear boundaries with wide margins. In Fourclass-2 with 3 uniformly distributed irrelevant features only J_2 and $\partial_{A,B}$ has succeeded. Fourclass-2+7M has 7 irrelevant multimodal features. As one can see from two-dimensional projections in Figure 40, those features can partially separate classes. Therefore, J_2 based on variance is confused by these additional features, same as all other measures except for $\partial_{A,B}$. $\partial_{A,B}$ captures density and local neighborhood, which were the most important characteristics in this case.

Birch-2+8 data set has 8 partially discriminative features, which are considered as irrelevant. Only J_2 and $\partial_{A,B}$ were able to see better class separability in just 2 relevant dimensions, where class boundaries are piecewise linear with wide margins.

Classes are heavily interleaved in all-relevant features versions of RBF-10, and RDG-10. RBF-10 looks somewhat like Gauss-8, only it has 50 centroids densely located, each of the centroids mimics a Gaussian distribution. The density is uneven and boundary is nonlinear. Though, contrary to Gauss-8, J_2 has shown increased class separability, as variances ratio in this case took effect. $\partial_{A,B}$ provided a good result on class separability, unlike all other measures.

RDG-10 looks like nearly uniformly distributed classes with no clear clusters or boundaries in relevant dimensions – the structure is barely visible (Figure 46). It provides a good illustration of what real structured data might look like when continuous features are used for rules of which decision lists for classification are made, only the natural process is reversed to create synthetic data. Yet this structure is detected by J_2 and $\partial_{A,B}$ in two relevant dimensions compared to irrelevant uniformly distributed ones.

Expecting improvement in class separability after elimination of irrelevant features, one should keep in mind that higher dimensions actually contribute to better class serapability, as shown in Table 8. This effect appears on a smaller scale (as can be seen from scaling down by the number of dimensions), so in most cases this improvement in class separability was insignificant in higher dimensions. $\partial_{A,B}$ has shown class separability improvement in all cases in spite of this effect taking place.

In order to obtain additional information on data structure we would like to evaluate geometrical complexity of classification problems using complexity

measures. Results are provided in Table 11. All cases where complexity is reduced as expected after elimination of irrelevant features are shown in bold. All complexity measures, except for T2, take their values in the interval (0..1), the smaller the less complex classification problem is.

TABLE 11 Classification problem complexity measured on synthetic and benchmark data sets.

Data	N1	N2	N3	T1	T2	L1	L2	L3
GaussS-2-all	**0.0008**	**0.2385**	0.0002	**0.7681**	833.17	**0.5448**	0.0002	0.0000
GaussS-2+1	**0.0004**	**0.0602**	0.0002	**0.2712**	1666.3	**0.5388**	0.0002	0.0000
GaussS-2+1M	**0.0004**	**0.0602**	0.0002	**0.2712**	1666.3	**0.5388**	0.0002	0.0000
GaussS-2+1B	**0.0010**	**0.0525**	**0.0006**	**0.2222**	1666.3	**0.5405**	0.0002	0.0000
GaussS-2+1U	**0.0004**	**0.0548**	0.0002	**0.2080**	1666.3	**0.5425**	0.0002	0.0000
GaussS-2	**0.0004**	**0.0130**	**0.0002**	**0.0600**	2499.5	**0.5382**	0.0002	0.0000
Gauss-8+10	**0.6521**	**0.4355**	**0.3605**	**1.0000**	277.72	0.9720	0.4715	0.4677
Gauss-8	**0.3147**	**0.1836**	**0.2463**	**0.9996**	624.87	0.9973	0.4999	0.5000
Foursubclass-2+10	**0.2492**	**0.1421**	0.2217	**1.0000**	83.25	0.9771	0.4815	0.4990
Foursubclass-2+5G	**0.0601**	**0.0300**	0.2442	**0.9950**	142.71	0.9929	0.4995	0.5000
Foursubclass-2+5U	**0.1772**	**0.0921**	0.2432	**1.0000**	142.71	0.9835	0.4965	**0.5010**
Foursubclass-2	**0.0290**	**0.0170**	0.2532	**0.4975**	499.50	0.9990	0.4995	**0.5000**
Clouds-2+10	**0.5807**	**0.4045**	**0.3316**	**1.0000**	416.58	0.7155	**0.2480**	**0.2092**
Clouds-2	**0.2198**	**0.1536**	**0.2928**	**0.8938**	2499.5	0.7162	**0.2470**	**0.2005**
Concentric-2+10	**0.5254**	**0.3313**	**0.2851**	**1.0000**	208.25	0.7373	**0.3685**	**0.5000**
Concentric-2	**0.1580**	**0.0918**	**0.0489**	**0.4930**	416.58	0.8807	**0.0196**	**0.0050**
Spirals-2+3	**0.5337**	**0.3252**	0.4145	**1.0000**	995.00	0.7733	0.3401	0.2913
Spirals-2	**0.1037**	**0.0466**	0.4172	**0.7992**	2487.5	0.7739	0.3405	0.2914
Fourclass-2+3	**0.1821**	**0.0853**	**0.1991**	**0.9850**	227.40	0.6111	0.2412	0.4837
Fourclass-2	**0.0044**	**0.0000**	**0.1896**	**0.3355**	568.50	0.6124	0.2414	0.4850
Fourclass-2+7M	**0.0172**	**0.0048**	0.0964	**0.9309**	126.33	0.5789	0.0754	0.1330
Fourclass-2	**0.0044**	**0.0000**	0.1896	**0.3355**	568.50	0.6124	0.2414	0.4850
Birch-2+8	**0.0023**	**0.0013**	**0.0282**	**0.8150**	1105.4	0.5238	0.0016	0.0102
Birch-2	**0.0013**	**0.0007**	**0.0248**	**0.2272**	5527.0	0.5240	0.0018	0.0111
RBF-10+10	**0.5921**	**0.3999**	**0.3132**	**1.0000**	249.95	**0.8454**	**0.3193**	0.3589
RBF-10	**0.1148**	**0.0616**	**0.2547**	**0.8778**	499.90	**0.8449**	**0.3187**	0.3657
RDG-10+10	**0.4883**	**0.3229**	0.1922	1.0000	249.95	**0.8087**	0.2282	**0.1896**
RDG-10	**0.3603**	**0.2164**	0.2110	1.0000	499.90	**0.8077**	0.2290	**0.1873**

Most of these measures are not informative individually as they cover different aspects of classification complexity. However, together they can supply additional information on data characteristics.

Out of 8 complexity measures evaluated on selected data sets N1, N2, and T1 appeared to be the most informative with respect to complexity created by

additional irrelevant features. All of them access proximity of instances from the opposite class based on the neighborhood concept.

L1 and L2 evaluate to what extent the classes are linearly separable. GaussS-2 is the only linearly separable case, but the margins are quite narrow. Birch-2 has a piecewise linear decision boundary with wide margins between classes. L1 and L2 in these cases were quite similar. With additional discriminative information misleadingly brought by irrelevant features, L2, the training error of a linear classifier, is quite low for Concentric-2+10 and Fourclass-2+7. T1 is a very informative measure when it comes to distinguishing cases with wide or narrow margins between classes, or if classes are intersecting.

Two measures, N2 and L3, evaluate error rates obtained by nonlinear and liner classifiers respectively on the data obtained using linear interpolation of the original data set. Error reduction after elimination of irrelevant features in both N2 and L3 are obtained for Clouds and Concentric data sets only, but in many cases from none of them. Considering properties of selected data sets, N3 predictably performed better.

Computational costs bring another important consideration for a class separability measure to be used as a criterion to optimize. In order to determine approximate computational costs for class separability measures, the following costs for basic operations are used, where M is a number of instances, $M = M_A+M_B$ for classes A and B, 2^L is the number of pairwise decompositions of L classes, N is the number of features:

- Mean vector for mean value in each feature, $O(MN)=O(M_AN)+O(M_BN)$;
- Variance vector for variance in each feature, $O(MN)=O(M_AN)+O(M_BN)$;
- Mean vectors difference, $O(N)$;
- Covariance matrix computation, $O(MN^2)$;
- Sum of N x N matrices, $O(N^2)$;
- N x N matrix multiplication, $O(N^3)$;
- N x 1 and 1 x N matrix multiplication, $O(N^2)$;
- 1 x N and N x N and N x 1 matrix multiplication, $O(N^2)+O(N)$;
- Matrix determinant (LU decomposition), $O(N^3)$;
- Matrix inversion by Gauss-Jordan elimination, $O(N^3)$;
- Matrix trace, $O(N)$;
- Covariance matrix inversion by Cholesky decomposition, $O(N^3/3+N^2/2)$;
- Counting discrete feature values in each class, $O(MN)$;
- Normalization of attribute values counts, $O(N)$;
- Calculation of probabilities, $O(0.5N\sqrt{M})$, where the number of feature values after unsupervised discretization is $0.5\sqrt{M}$;
- Distances between all pairs of instances from different classes, $O(M_AM_BN)$;
- Distances between all pairs of instances of the same class, $O(M_A(M_A\text{-}1)N+M_B(M_B\text{-}1)N))$;
- Finding a nearest neighbor of the opposite class in one go, $O(2M_AM_B)$;
- Finding a nearest neighbor of the same class in one go, $O(M_A(M_A\text{-}1)+M_B(M_B\text{-}1))$;

Operations for transposing a matrix and logarithm computations are not taken into account as well as elementary mathematical operations of O(1).

Approximate computational costs for two-class problems are derived as follows.

$NIR_{A,B}$: O(MN^2)+O(N^2)+O(N^3)+O(MN)+O(N)+O(N^2)+2O(N^3)=
=O(N(M+1)+N^2(M+2)+3N^3);

$a_{A,B}$: O(MN)+O(N)+O(MN^2)+O(N^2)+O(N^2)+O(N)+3O(N^3)=
=O(N(M+2)+N^2(M+2)+3N^3);

$KL_{A,B}$, $DIV_{A,B}$: O(MN)+O(N)+O(0.5$N\sqrt{M}$)=O(N(M+1+0.5$\sqrt{M}$));

J_1: 2O(MN^2)+O(N^2)+O(N^3)+O(MN)+3O(N)+2O(N^2)+O(N^2)+O(N^3)+O(N)=
=O((MN)(2N+1)+4N(1+N)+2N^3);

J_2: 2O(MN^2)+O(N^2)+O(MN)+3O(N)+2O(N^2)+O(N^2)+2O(N)=
O((MN)(2N+1)+5N+4N^2);

$F_{A,B}$: O(MN)+O(MN)=O(2MN);

$\partial_{A,B}$: 2O($M_A M_B N$)+O(M_A(M_A-1)N+M_B(M_B-1)N))+2O($M_A M_B$)+O(M_A(M_A-1)+M_B(M_B-1))=O((2$M_A M_B$+M_A(M_A-1)+M_B(M_B-1))(N+1));

Alternatively, $\partial_{A,B}$: O(M^2N).

Computational costs of class separability and other complexity measures are calculated using a difference in one feature between two instances as a basic operation, d. Computational costs estimates are provided in Table 12 taking M as the number of instances in the training set, N as the number of features, and L as the number of classes. Using the expressions above rough estimates for computational costs for all class separability measures are made. These estimates provide the upper cost limit, but they are satisfactory to support the conclusions on class separability measures application: $\partial_{A,B}$ is one of the most computationally intensive measures. However, much less computations are used in BDP implementation, as only intra-class distance is minimized via neighborhood variances minimization.

TABLE 12 Approximate computational costs for 1000 instances, 2 classes, 500 each class, 10 features.

Class separability	$NIR_{A,B}$	$a_{A,B}$	$KL_{A,B}$	$DIV_{A,B}$	J_1	J_2	$F_{A,B}$	$\partial_{A,B}$
Computational costs	113210	113220	10170	10170	212440	210450	20000	10989000

Approximate computational complexity of weights adaptation in one run is O((M_A(M_A-1)+M_B(M_B-1))(N+1)). It breaks down computational complexity of $\partial_{A,B}$ measure by half. There are many ways to simplify computation of these measures described in the literature, which can be used to improve BDP in the future.

6.2 Investigation of superclass / subclass structure

Class heterogeneity in data means that there exists a superclass/subclass structure beyond class labels. In other words, class distribution can be a mixture distribution having a few modes. Statistically, it is not necessarily recognized as a multimodal distribution. Such techniques as, for example, Linear Discriminant Analysis and Kernel Based Analysis assume that each class has a single Gaussian distribution in the original or transformed feature space (You & Martinez, 2010). This assumption is too restrictive and seldom met in practice. In order to relax this assumption, single class can be viewed as a mixture of Gaussians (You & Martinez, 2010). A similar reasoning is applied in bidirectional partitioning. In this section, synthetic and benchmark data sets are used to study performance of BDP in case of class heterogeneity exhibited in low dimensions.

6.2.1 A case study of distance-based grouping on binary data

The crucial part of finding local groups of instances in subspaces with improved class discrimination is distance-based grouping that uses obtained pairwise feature weights. The existing data structure is changed as distances are changed by weights. It is expected that distances between groups will become considerably larger than distances between different classes within a group after agglomerative merging. In addition, as the class separability within a group increases, instances of the same class become relocated densely. We verify this assumption using a simple synthetic example. Let us consider a manually computable example with predefined groups and pre-set weights (Table 13).

Example 1 includes 2 groups, 6 binary features, 12 instances – 6 in each group, and 2 classes – 3 instances of each class in each group. Features f_1, f_2, and f_3 are relevant for the first group and take uniformly distributed random values in the second group; features f_4, f_5, and f_6 are relevant for the second group and take uniformly distributed random values in the first group.

Weight adaptation is performed through a pre-defined number of iterations. Initially all weights are equal. Local neighborhood is found using weighted Manhattan distance, not a weighted exponential distance. Local neighborhood for each instance may include instances of the same and a different group. Our goal is to have at least majority of same-group instances in the neighborhood. In the synthetic example, during first iteration of weights adaptation instances #12, 14, and 19 had minority of instances of a different group in their neighborhood. In second iteration due to weights adaptation all instances have only same-group nearest neighbors.

Some irrelevant features have small variance in the local neighborhood and obtain high weight. This may introduce an error at computation of pairwise weights and distances. This error is corrected by choosing maximal weight of two single weights for a pairwise weight in case of IPA>0.5 between single

weight vectors as $w_{r,s}^{j} = \max(w_r^j, w_s^j)$ and minimal of two single weights otherwise. In order to enhance the effect, all weighted distances are multiplied by (IPA + 1), where IPA is measured between weight vectors. Therefore, pairwise weights do not sum up to 1 in all features on the instance pairs from different groups. Table 14 demonstrates distribution of weights after 5 iterations in our synthetic example. With the above weights weighted inverse exponential distances are distributed as follows: distances between instances of the same class, same group appear in the interval [0.001…0.016], distances between instances of the same class and different group appear in the interval [0.112…0.303]. These distances are given as an input to a distance-based clustering algorithm. DBSCAN implemented in WEKA (Hall *et al.,* 2009) has been used. In our example, DBSCAN with a radius 0.0102 assigns instances #0-5 to the subgroup #1 and instances #12-18 to the subgroup #3. Instances #6-11 and #19-25 are assigned to the subgroups #2 and #4 accordingly.

TABLE 13 A simple illustrative binary data example of feature space heterogeneity.

#	f_1	f_2	f_3	f_4	f_5	f_6	Class	Group
0	**0**	**1**	**0**	0	0	0	0	0
1	**0**	**1**	**0**	0	1	0	0	0
2	**0**	**1**	**0**	0	1	1	0	0
3	**0**	**1**	**0**	1	0	0	0	0
4	**0**	**1**	**0**	1	0	1	0	0
5	**0**	**1**	**0**	1	1	1	0	0
6	**0**	**1**	**0**	0	0	0	1	0
7	**0**	**1**	**0**	0	1	0	1	0
8	**0**	**1**	**0**	0	1	1	1	0
9	**0**	**1**	**0**	1	0	0	1	0
10	**0**	**1**	**0**	1	0	1	1	0
11	**0**	**1**	**0**	1	1	1	1	0
12	0	0	0	**1**	**1**	**0**	0	1
13	0	0	1	**1**	**1**	**0**	0	1
14	0	1	1	**1**	**1**	**0**	0	1
15	1	0	0	**1**	**1**	**0**	0	1
16	1	1	0	**1**	**1**	**0**	0	1
17	1	1	1	**1**	**1**	**0**	0	1
18	0	0	0	**1**	**1**	**0**	1	1
19	0	0	1	**1**	**1**	**0**	1	1
20	0	1	1	**1**	**1**	**0**	1	1
21	1	0	0	**1**	**1**	**0**	1	1
22	1	1	0	**1**	**1**	**0**	1	1
23	1	1	1	**1**	**1**	**0**	1	1

At this step, parameter tuning for radius is required in order to correctly identify one-class group components. Among all parameters, the most important are those that define the local neighborhood, because no default value can be used in this case. Proportion of the same-class and different-class instances in the local neighborhood is crucial for weight adjustments and subsequent weighted distance-based grouping. In order to preserve the data structure and produce correct estimates with respect to varying density of instances, it is suggested to

use pre-estimated ε radius and where appropriate, set the number of nearest neighbors dynamically using this radius. In our implementation we used a pre-estimated empirical value for the DBSCAN radius parameter which is an average distance to the k^{th} neighbor of the same class.

TABLE 14 Single feature weights after 5 iterations.

#	w_1	w_2	w_3	w_4	w_5	w_6
0	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
1	**0.314**	**0.314**	**0.314**	0.026	0.007	0.026
2	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
3	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
4	**0.314**	**0.314**	**0.314**	0.026	0.026	0.007
5	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
6	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
7	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
8	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
9	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
10	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
11	**0.308**	**0.308**	**0.308**	0.025	0.025	0.025
12	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
13	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
14	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
15	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
16	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
17	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
18	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
19	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
20	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
21	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
22	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
23	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
24	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**
25	0.025	0.025	0.025	**0.308**	**0.308**	**0.308**

In this case, DBCSAN was able to find subclasses and assign them to different subgroups. Every subgroup contains instances of one class given the distance distribution regulated by β=2. At the next step, subgroups are joined into groups. Subgroups #1 and #2, #3 and #4 are joined with IPA=0, that is they have identical weights distribution. Weighted distances were given as an input for DBSCAN in Weka (Hall *et al.*, 2009). DBSCAN clusters instances disregarding their class labels. DBSCAN with radius $\varepsilon \in$ [0.33…0.66] assigns instances #7, 8, and 9 to one group and the rest to the other group, with radius $\varepsilon \in$ [0.1…0.32] DBSCAN produces 4 clusters, each includes 3 instances of the same class and same group. The same experiment has been performed with COSA original software and average-linkage hierarchical clustering. Instance #1 was assigned to the incorrect group. With the correct radius threshold, all found groups included instances of the same class and same known group, otherwise DBSCAN as a grouping method did not produce correct group assignments. Irrelevant features have a negative impact, but even with completely irrelevant features one-class group components were identified correctly.

There is a tradeoff between side effects of using the feature-specific radius components and radius obtained over all features. On one hand, the interactive features, which are discriminative only when considered together, may be underestimated. On the other hand, the neighborhood defined in the original feature space will be affected by all irrelevant features. In this case, we follow the assumption of feature independence. Results of weighted inverse exponential distance computations given as input to a clustering algorithms are shown in Table 15.

TABLE 15 Weighted distances matrix in Bidirectional Data Partitioning (BDP) for all pairs created out of 12 instances. Not highlighted main diagonal shows distances between instances of the same class that belong to the same group. Two other diagonals along the main one present the distances for different classes that belong to the same group. The anti-diagonal shows distances of different classes that belong to different groups. Two other diagonals along the anti-diagonal are distances between instances of the same class that belong to different groups. Cases that were not assigned to the correct subgroup by a clustering algorithm are shown in italic.

	1	2	3	4	5	6	7	8	9	10	11	12
1	-	0.03	0.08	-0.91	-0.96	-0.91	0.21	0.21	0.13	-0.86	-0.78	-0.91
2	0.03	-	0.03	-0.96	-0.91	-0.91	0.35	0.35	0.21	0.78	0.64	0.86
3	0.08	0.03	-	-0.91	-0.96	-0.86	0.21	0.21	0.13	-0.86	-0.78	-0.91
4	-0.91	-0.96	-0.91	-	*0.13*	0.03	-0.78	-0.78	-0.86	0.13	0.21	0.08
5	-0.96	-0.91	-0.91	*0.13*	-	0.08	-0.86	-0.86	-0.91	0.08	0.13	0.03
6	-0.91	-0.96	-0.86	0.03	0.08	-	-0.86	-0.86	-0.91	0.08	0.13	0.03
7	0.21	0.21	0.13	-0.78	-0.78	-0.86	-	0.08	0.03	-0.96	-0.91	-0.91
8	0.35	0.35	0.21	-0.86	-0.86	-0.91	0.08	-	0.03	-0.96	-1.00	-0.91
9	0.21	0.21	0.13	-0.86	-0.86	-0.91	0.03	0.03	-	-1.00	-0.96	-0.96
10	-0.86	-0.78	-0.91	0.13	0.21	*0.08*	-0.96	-0.91	-0.91	-	0.03	0.03
11	0.78	0.64	0.86	*0.08*	0.13	*0.03*	-0.96	-1.00	-0.91	0.03	-	0.03
12	-0.86	-0.78	0.91	*0.08*	0.13	*0.03*	-1.00	-0.96	-0.96	0.03	0.08	-

In the next step, pure class subgroups are combined into final groups based on the agglomerative merging procedure that uses feature weights in calculation of the Importance Profile Angle (IPA). Weight profiles are obtained by finding median or average feature weight in each group, w_l^j. After a few iterations those weights stabilize and should have relatively close values within a group compared to the other groups. Feature weights profiles $\left(w_1^1, \ldots, w_l^j, \ldots, w_L^N\right)$ are found for each subgroup and IPA between these vectors is found for all pairs of subgroups. At each step, two groups with smallest IPA are merged, if their IPA does not exceed an IPA threshold (empirical value is used, commonly in the interval [0.2…0.5]).

6.2.2 Experiments with class decomposition and two classifier combination schemes

The aim of this experimental series is to investigate performance of two decomposition schemes: (1) with meta-classifier that uses cluster labels as new

class labels; (2) with IPA-based agglomerative grouping procedure and distance-based component classifier selection. As an illustrative example of BDP technique we shell use a synthetic data set that resembles Fourclass-2, but has wide margins (Figure 29). Subclass 0 of class c0 in the lower left is created as G(5.0; 0.25) in features f_1 and f_2. Subclass 0 of class c1 in the lower right has G(10.0;0.25) in f_1 and G(5.0; 0.25) in f_2. Subclass 1 of class c0 in the upper right has G(10.0; 0.25) in both f_1 and f_2. Subclass 1 of class c1 in the upper left has G(5.0;0.25) in f_1 and G(10.0; 0.25) in f_2. Original version before normalization is used to facilitate results interpretation.

Class separability measurements for this data are:

$NIR_{A,B}$ – 1.1636; $a_{A,B}$ - 1.9546; $KL_{A,B}$, $DIV_{A,B}$ = 0.0; J_1 = 0.0; J_2 = 975560.7614; $F_{A,B}$ = 0.0; $\partial_{A,B}$ = 0.0047. T1 characterizes complexity of class boundary with the value 0.9722, which signifies rather high complexity (the ratio threshold is fixed). Fraction of points on the class boundary, N1, is 0.0015, which is reasonable for wide margins between classes. A complexity measure that characterizes linearity of decision boundary, L1, correctly assigns this data set a value 0.9997, which corresponds to a non-linear boundary. Other complexity measures that evaluate error of a linear model, L2, and non-linearity of a liner model, L3, are approximately 0.5.

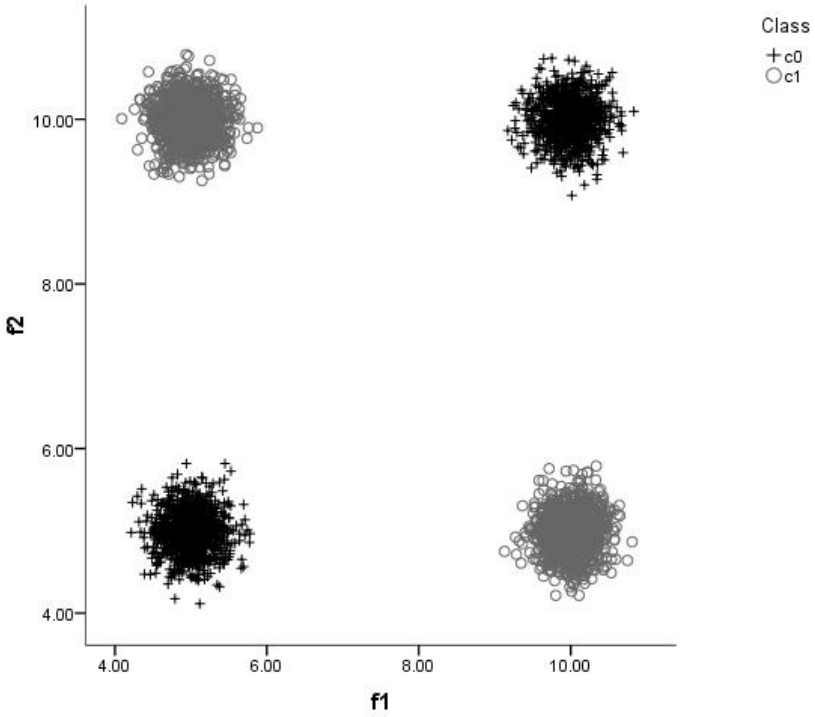

FIGURE 12 A synthetic data set with two classes, two subclasses each, Foursubclass-2.

Nearly all measures consider this case as poor class separability, except for $\partial_{A,B}$ (J_2 and $\partial_{A,B}$ has small values in case of good class separability, unlike other measures). From the point of evaluation function, poor separability

characteristic is fine, because this data set needs decomposition in order to improve performance of most classifiers as will be shown below. There are a few exceptions, for example, *k*-NN and LibSVM Support Vector Machine (SVM) with polynomial kernel that are 100% accurate. For that kind of classifiers $\partial_{A,B}$ as an evaluation function would tell BDP to stop partitioning.

In general, the best performance would be achieved if this data set is partitioned so that one model is built per pair of subclasses that belong to different classes. However, if any three subclasses are assigned to one group and the fourth subclass is assigned to another group, the local models are still linearly separable. Performance will depend on how wide the margins between subclasses are and how a particular base classifier treats that kind of margins.

First, let us consider an example of BDP that updates weights through the entire data set and subsequently clusters instances using weighted clustering, in this example – DBSCAN. Both features are relevant; hence, weights do not change dramatically. However, small change in feature weights entails different clustering results compared to distance-based clustering with equal feature weights.

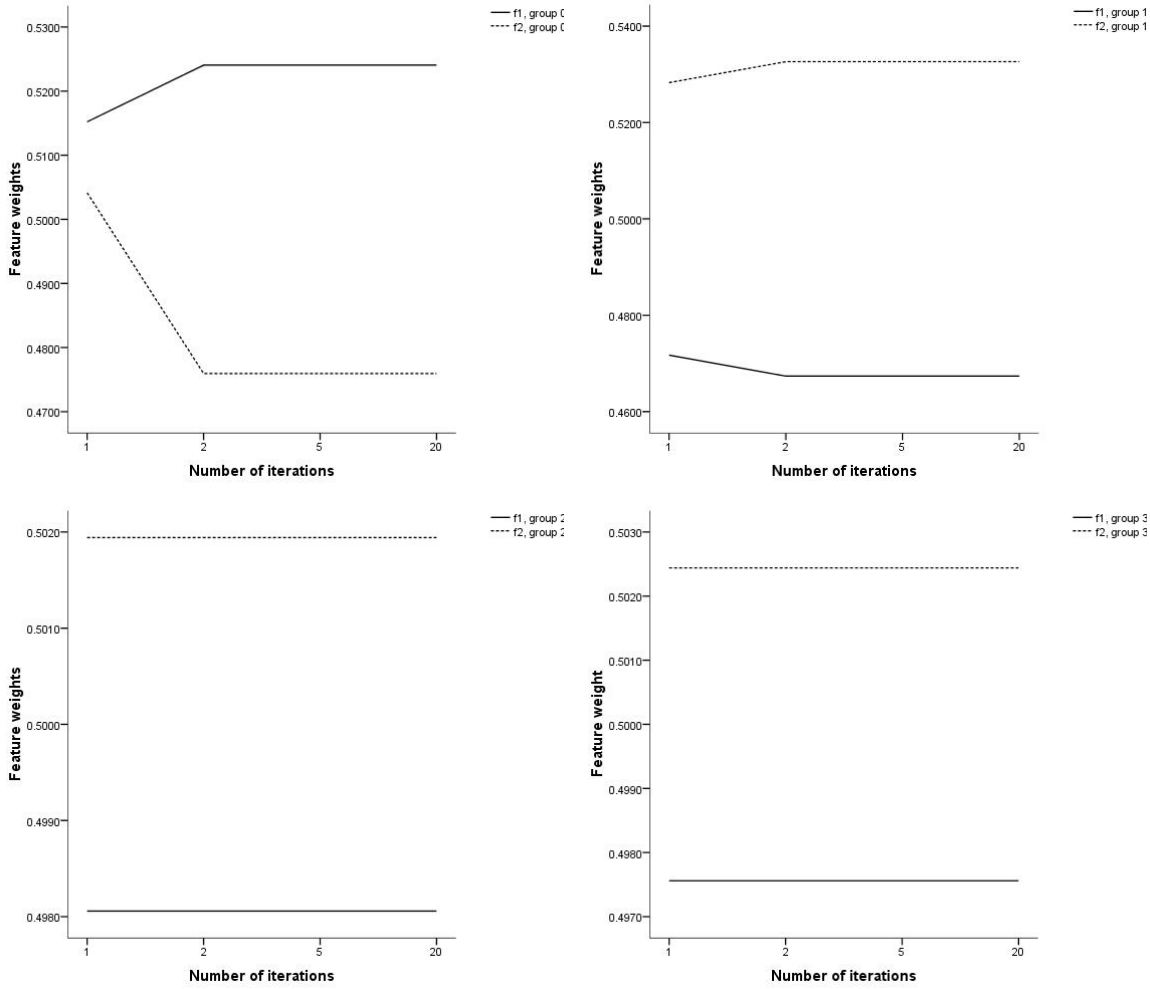

FIGURE 13 Feature weights in four subgroups of Foursubclass-2 data set. Each subgroup of instances found by weighted clustering corresponds to a subclass.

The results below also demonstrate optional output during model construction in BDP software. IPA threshold for grouping is 0.002. Feature weights after 2 iterations:

Median weight of feature #0 in the group #0: 0.5016
Median weight of feature #1 in the group #0: 0.4984
Median weight of feature #0 in the group #1: 0.4972
Median weight of feature #1 in the group #1: 0.5028
Median weight of feature #0 in the group #2: 0.4983
Median weight of feature #1 in the group #2: 0.5017
Median weight of feature #0 in the group #3: 0.5008
Median weight of feature #1 in the group #3: 0.4992

Feature weights after 20 iterations

Median weight of feature #0 in the group #0: 0.5010
Median weight of feature #1 in the group #0: 0.4990
Median weight of feature #0 in the group #1: 0.5022
Median weight of feature #1 in the group #1: 0.4978
Median weight of feature #0 in the group #2: 0.4995
Median weight of feature #1 in the group #2: 0.5005
Median weight of feature #0 in the group #3: 0.5026
Median weight of feature #1 in the group #3: 0.4974

We can see that weights have stabilized after only 2 iterations (Figure 13). Below are the results of agglomerative procedure of subgroups merging based on IPA.

Round 1.

IPA, group #0 and group #1 : 0.0015
IPA, group #0 and group #2 : 0.0019
IPA, group #0 and group #3 : 0.0020
IPA, group #1 and group #0 : 0.0015
IPA, group #1 and group #2 : 0.0034
IPA, group #1 and group #3 : 0.0005
IPA, group #2 and group #0 : 0.0019
IPA, group #2 and group #1 : 0.0034
IPA, group #2 and group #3 : 0.0039
IPA, group #3 and group #0 : 0.0020
IPA, group #3 and group #1 : 0.0005
IPA, group #3 and group #2 : 0.0039
Lowest IPA, group #1 and group #3 : 0.0005

Round 2.

IPA, group #0 and group #1 : 0.0020
IPA, group #0 and group #2 : 0.0019
IPA, group #1 and group #0 : 0.0020
IPA, group #1 and group #2 : 0.0039
IPA, group #2 and group #0 : 0.0019
IPA, group #2 and group #1 : 0.0039
Lowest IPA, group #0 and group #2 : 0.0019

Round 3.

IPA, group #0 and group #1 : 0.0020
IPA, group #1 and group #0 : 0.0020
Lowest IPA, group #0 and group #1 : 0.0020
But these groups will not be merged, because IPA $\geq$ 0.002.

Total number of instances in group #0 = 1920, class distribution:
class 0: 954 instances,
class 1: 966 instances.
Total number of instances in group #1 = 1911, class distribution:
class 0: 945 instances,
class 1: 966 instances.

A relatively small number of instances are marked as "noise" by DBSCAN procedure and do not participate in model construction. If an IPA threshold value is small enough, the final partitioning would be the same as in original data, and BDP results would be the same as the result for a basic classifier used with it.

IPA agglomerative merging procedure is based on feature weights. In case weights adaptation is not performed, either all or none of subgroups are joined, depending on the threshold. If each subgroup is used to build a model, all component classifiers are based on a simple one-class rule. Integration of the component classifiers to find the right model is based on a *k*-NN procedure; hence, BDP performance is similar to *k*-NN classifier that uses weighted inverse exponential distance[1].

DBSCAN has two parameters, (1) minimum number of instances required in an epsilon-range-query, minPoints = $k/2$, and (2) radius of the epsilon-range-queries, □, set to an average distance to the k^{th} neighbor of the same class, where distance is weighted inverse exponential and k = 0.5* $\sqrt{M}$, M is a training set size.

For the Foursubclass-2 data set, DBSCAN with these parameters returns the same results with or without weights adaptation: every subclass assigned to its own subgroup for subsequent merging, if chosen. However, *k*-Means produces different results with and without weights having the same number of clusters as a parameter.

When clustering inside classes (CIC) is chosen as an option, clustering is performed for each class separately and subgroups from each class are collected for further merging. In case DBSCAN applied to Foursubclass-2 derives the same partitioning, CIC or not, with weights adaptation or without.

Alternatively, integration of the component classifiers can be performed with meta-classifier that uses cluster labels as new class labels, that is each subgroup of instances is used to build a separate model.

---

[1] BDP is functioning as *k*-NN with weighted inverse exponential distance, if clustering inside classes is chosen along with IPA threshold set to -1 (outside the possible range [0..1]), and weight adaptation is performed. BDP is functioning as a base classifier used as a global model, if IPA threshold is more than 1.

WEKA's Multi-Class Classifier is not able to improve any of those poor-performing base algorithms, neither is AdaBoostM1. Bagging is able to raise performance of J48 to 98.6% but no other algorithm's performance. Among variety of base algorithms and ensemble schemes tests, Ridor tree-based rule learner (98.0%) and RandomTree (98.5%) with random selection of features at each node are noted for high accuracy on this data set. It confirms that this kind of classification problem is difficult for most classifiers and ensemble techniques of which WEKA has a good representation.

6.2.3 Evaluation of different BDP schemes on benchmark data

Usually, the presence of subclasses and/or super classes in data is unknown, but in some cases domain knowledge suggests a potential ability to discover them. Therefore, a few benchmark data sets have been used. In these experiments, data partitioning has been performed at different levels of granularity according to different strategies.

Wine data set (Blake *et al.*, 1998) has linearly separable classes. There are 13 continuous features and 3 classes. Class separability measurements for this data set are:

$NIR_{A,B}$ – 9.1320, $a_{A,B}$ – 8.8791, $KL_{A,B}$ – 0.1087, $DIV_{A,B}$ – 0.2175, J_1 – 13.8701, J_2 – 2.6254, $F_{A,B}$ – 329.7224, $\partial_{A,B}$ – 0.0558 (Manhattan distance-based). The most informative measures, J_2 and $\partial_{A,B}$, suggest good class separability for this data set.

Out of different BDP schemes evaluated, the most interesting results are presented in Table 16. All BDP schemes use weight adaptation performed in 5 iterations, pre-estimation of radius parameter for DBSCAN performed on the entire data set. The base classifiers used in this experiment are J48 pruned decision tree, 1-Nearest Neighbor (IB1), and NaiveBayes with Gaussian distribution estimator (NB). No feature selection has been performed. Preliminary experiments have confirmed that feature selection has no benefit in accuracy in case of J48, because it has an embedded feature selection. The BDP schemes used are:

1. BDP with clustering on entire data set (noCIC) using DBSCAN, IPA merging with manually selected threshold (user-specified), weighted inverse exponential distance based integration of component classifiers based on nearest group (WIED-NN) (BDP-1 in Table 7, Subsection 5.3.3).
2. BDP with clustering inside classes (CIC) using DBSCAN, IPA merging with manually selected threshold, WIED-NN component classifiers integration (BDP-2).
3. BDP with clustering on entire data set (noCIC) using *k*-Means with manually selected number of clusters per class, IPA merging with manually selected threshold (user-specified), WIED-NN component classifiers integration (BDP-3).
4. BDP with clustering inside classes (CIC) using *k*-Means with manually selected number of clusters per class, IPA merging with manually selected threshold (user-specified), WIED-NN component classifiers integration

(BDP-4).

5. BDP with clustering inside classes (CIC) using *k*-Means with manually selected number of clusters per class, component classifiers integration using meta-classifier that maps clusters to classes (MetaC2C) (BDP-8).

TABLE 16 Wine benchmark data set: Bidirectional Data Partitioning (BDP) with weight adaptation vs. Multi-Class Classifier (MCC) with pairwise class combination. Pruned J48 decision tree is used as a base classifier. Additional characteristics of BDP scheme are provided. In particular, pre-set threshold values (number of clusters in *k*-Means, IPA-threshold for merging subgroups) and pre-estimated threshold values (□-radius for DBSCAN and number of obtained clusters in DBSCAN). The number of clusters, # clusters, is given per one class, multiplied by 3 for 3 classes. In case *k*-Means was not able to find pre-set number of clusters, the actual number of clusters is given in parentheses. The number of final groups after IPA-merging, if different, is provided in square brackets. Accuracy is measured using 10-folds CVM performed in a single run.

BDP parameters and performance	BDP-1	BDP-1	BDP-3	BDP-4	BDP-4	BDP-8	BDP-8	Base classi-fier	MCC Pairwise
CIC	no	no	no	yes	yes	yes	yes	-	-
Clusterer	DBSCAN	DBSCAN	*k*-Means	*k*-Means	*k*-Means	*k*-Means	*k*-Means	-	-
# clusters	3[2]	3	18(4)[4]	6x3[14]	6x3[17]	4x3[12]	6x3[18]	-	-
□-radius	0.09	0.09	-	-	-	-	-	-	-
IPA	0.2	0.05	0.05	0.1	0.05	-	-	-	-
Integration	WIED-NN	WIED-NN	WIED-NN	WIED-NN	WIED-NN	MetaC2C	MetaC2C	-	-
Acc, %, J48	94.9438	98.3146	89.3258	96.0674	96.6292	91.0112	93.8202	93.8202	89.8876
Acc, %, IB1	96.0674	98.3146	93.2584	95.5056	96.6292	94.9438	94.9438	94.9438	94.3820
Acc, %, NB	96.6292	98.3146	97.7528	96.0674	96.6292	98.3146	98.3146	96.6292	97.1910

The highest accuracy level with all classifiers has been achieved using BDP-1 with 3 clusters. In this case BDP produced groups of instances each consisting of one class only, and the base classifier is a very simple rule assigning a test instance to that particular class. Therefore, BDP performed as a weighted 1-Nearest Neighbor. It outperforms unweighted 1-Nearest Neighbor by approximately 4%. Somewhat lower accuracy level (94.9%) has been achieved using BDP-1 with IPA threshold = 0.2, which merged 2 of three classes into one group. BDP-4 that created 6 clusters per class using *k*-Means clustering. Joining two of them with IPA threshold = 0.05 lead to highest accuracy level with 17 component classifiers compared to 14 classifiers using distance-based integration.

k-Means works better if clustering is performed inside classes having the same total number of classes. *k*-Means performs better with IPA joining than with meta-classifier.

DBSCAN cannot find clusters inside classes, either used with global or inside class pre-estimation of radius. Best performance achieved if every class forms a group with no FS, than BDP works as a weighted nearest neighbor.

The highest accuracy achieved for Wine data set corresponds to 3 misclassified instances, one instance from one class, and two instances from another. BDP results suggest that one class is better separable from the other two. This interpretation is hard to obtain just from two-dimensional data projections out of 14 dimensions. In order to verify this result, class separability has been calculated in one-against-all and pairwise class combinations.

1-against-2&3:
- J_2: 0.4749
- $\partial_{A,B}$: 0.0554

2-against-1&3:
- J_2: 1.8980
- $\partial_{A,B}$: 0.0537

3-against-1&2:
- J_2: 18.6887
- $\partial_{A,B}$: 0.0539

1-vs-2:
- J_2: 0.4082
- $\partial_{A,B}$: 0.0597

1-vs-3:
- J_2: 0.5672
- $\partial_{A,B}$: 0.0534

2-vs-3:
- J_2: 6.9008
- $\partial_{A,B}$: 0.0579

These results suggest that class 1 is easier to separate than classes 2 and 3. There is limited domain knowledge on this data set, but it is known that in the original study classes have been described using 30 features. A new study on wine characteristics from different origins might suggest what factors influenced similarity of origin 2 and 3, and what stays behind 6 per class groups discovered by BDP with *k*-Means that contributes to dissimilarity in wine characteristics.

All the above BDP schemes with local feature selection performed by CFS decreased accuracy, because CFS tends to select small number of features. Local feature selection by ReliefF and Information Gain using 4 features as cut-value were not able to raise the classification accuracy. Feature selection by means of feature weights with a median weight as a cut-value was not able to increase accuracy either for this particular data set. In case BDP has one-class groups and functions as a weighted 1-Nearest Neighbor, local feature selection cannot be performed.

6.2.4 Evaluation of BDP with correlation-based feature selection

In this section, experiments with weight adaptation are presented on continuous and binary data. In the data set syn_1 features f_1, f_2, f_3 are relevant, features f_4, f_5, f_6, f_7, f_8, f_9 are locally relevant, features f_{10}, f_{11}, f_{12} are globally irrelevant. Features take binary values. Relevant features are generated so that they take a certain value within a class with a probability of 0.99, and another value with a probability 0.01. Globally relevant features follow this distribution on 400 instances of the original data. Irrelevant features take both "0" and "1" values with probability equal to 0.5 on 400 instances of the original data.

The data set has two groups, first one includes instances from 0 to 199, second includes instances from 200 to 399. Each group consists of two classes, where the first 100 instances belong to class "0" and the second 100 instances belong to class "1". Features f_4, f_5, and f_6 are relevant for the first group of instances, and features f_7, f_8, and f_9 are relevant for the second group of instances. The data set has been split randomly onto training and test sets with probability 0.60 to be included to the training set and probability 0.40 to be included to the test set. As a result, the training set includes 240 instances, 17 of which are duplicate cases. The test set includes 160 instances, 9 of which are duplicate cases. Among duplicate cases there were inconsistency – the same instances belong to different classes and groups. A special filter RemoveInconsistant has been implemented in WEKA for this case study to remove these instances as a part of pre-processing step. Results of comparing BDP and one-against-all class encoding are shown in Table 17.

TABLE 17 Bidirectional Data Partitioning (BDP) with correlation-based feature selection (CFS) vs. Multi-Class Classifier (MCC) on synthetic binary data sets with different width of classes intersection interval. In BDP, clustering has been performed on entire data set using DBSCAN, integration has been performed using distance-based component classifier selection, feature selection in subproblems has been performed using CFS, and J48 has been used as a base classifier.

Data set	Classification accuracy, %.	
	BDP+CFS	MCC
Syn_bin_0.99_no_irr	99.50	96.50
Syn_bin_0.85_no_irr	90.25	79.25
Syn_bin_0.75_no_irr	69.75	62.00
Syn_bin_0.99_10_irr	99.25	96.50
Syn_bin_0.85_10_irr	90.75	79.25
Syn_bin_0.75_10_irr	73.50	70.00

As can be seen from Table 17, BDP outperforms Multi-Class Classifier (MCC). Correlation-based feature selection was able to identify irrelevant features

locally in most cases, therefore, accuracy with and without irrelevant features is nearly the same. Overlap between classes affected performance of both BDP and MCC.

6.3 Chapter summary

In this chapter we have investigated various aspects of Bidirectional Data Partitioning (BDP) technique design empirically on synthetic and benchmark data sets. Various class separability measures have been evaluated on data sets of different geometrical complexity with the following properties of interest: (1) Gaussian or non-Gaussian classes; (2) classes of even or uneven density; (3) wide, narrow margins between classes or completely interleaved classes; (4) subclasses or no subclasses; (5) linear or non-linear class separability; (6) irrelevant features as unimodal, multimodal, uniform, or mixed distributions that statistically cannot be qualified as multimodal; (7) different number of dimensions per irrelevant features; (8) two-class or multi-class problems.

Irrelevant features are independent of the class variable in reality, but incidentally, they may still exhibit a non-zero correlation with the class. Such co-incidences influence results on class separability obtained on synthetic and benchmark data sets. Nevertheless, intra- and interclass ratio as a measure of class separability ($\partial_{A,B}$) appeared to be less sensitive to this problem. It gave the best results out of eight class separability measures compared, but it's computationally the most demanding. Another measure that demonstrated good results is a scatter-matrix based measure (J_2) that is a ratio of variances (sum of within-class variances in all features to the sum of total variances in all features). This measure is also closely related to the one implemented in BDP for weight adaptation that minimizes variance of features within a group.

Intra- and inter-class ratio does not make any assumption regarding class distribution, does not rely on means and covariances, and does not require balanced classes. It depends on the distance function being based on a neighborhood concept. Related research on distance metrics has established the best distance function for high-dimensional applications is Mahalanobis distance (Aggarval *et al.*, 2001).

Implementation of BDP is based on heuristic search strategy to improve intra- and inter-class distances for a class separabiliy criterion, specifically, to reduce intra-class distance in multi-class problems. This heuristic has been tested on synthetic and benchmark problems with a subclass structure, that is class heterogeneity. Same as the intra- and inter-class ratio measure was the most effective in presence of subclasses, BDP technique was also effective in case of class heterogeneity.

Weight adaptation in BDP is able to detect noisy, missing, and irrelevant features. This is confirmed by numerous experiments on benchmark data sets with noise infusion and known irrelevant features.

Experiments on benchmark data sets confirm that data partitioning performed in BDP also reduces complexity in classification subproblems according to the best complexity measures established in our experiments: fraction of points on the class boundary (N1), the leave-one-out error rate of 1-NN classifier (N2), and the fraction of maximum covering spheres (T1).

As BDP builds local models based on data partitioning results, those models are combined under ensemble framework. Two classifiers integration schemes were studied: selection using a meta-classifier and distance-based selection. The experiments show that while meta-classifier shows better results in most cases, in particular situations distance-based selection is preferable.

BDP local models are built on unbalanced groups. It decreases performance. Imbalanced class representation is a common problem in data mining. Data generation or resampling using the Synthetic Minority Oversampling Technique (SMOTE) are possible solutions (Chawla *et al*., 2002).

Clustering techniques that are used and potentially can be used in BDP all have different characteristics leading to success in different situations. For example, DBSCAN can identify clusters of arbitrary shapes, but cannot handle clusters of uneven density. Parameter tuning in DBSCAN is a very computationally demanding task. The results vary drastically, yet the only way to verify them is classification accuracy of BDP, that is computationally demanding too.

7 EXPERIMENTAL STUDY

This chapter describes an empirical evaluation of the suggested decomposition strategies for class heterogeneity and feature space heterogeneity. Related data pre-processing topics are covered in the beginning. Decompositions by means of class encoding and decomposition with an IPA tree are studied. SEER Cancer data are used for evaluation of the suggested decomposition approaches. The chapter is summarized with conclusions.

7.1 Data pre-processing and exploratory analysis techniques

Data pre-processing is a very important step in data mining to prepare data for further analysis. A number of issues with data arise when it comes to application of a predictive technique. Some algorithms have built-in discretization, normalization, dealing with missing values, etc., while other algorithms have not. Data analysis without pre-processing may lead to inefficient model construction, or even entail incorrect results. For example, a particular variable may have numeric values, and in addition, some categorical values encoded by numbers. Thus, it has to be encoded at the pre-processing step. In addition, data may have imbalanced class representation, missing class labels, inconsistent data (for example, similar instances that belong to different classes), differences in scales of measuring features, and different feature types.

Among commonly used data-preprocessing techniques are normalization, standardization, discretization, encoding, and re-sampling. Usually, feature transformation techniques, such as feature selection and feature extraction, are

not considered a part of data pre-processing. Therefore, these topics are not covered here.

This section briefly reviews techniques relevant to the experimental study, feature selection and classification methods used, and also pays attention to selection of a distance function as a similarity measure.

7.1.1 Normalization, standardization, and discretization

Most techniques considered in this thesis are designed to deal with discrete numeric or nominal features. Many learning algorithms require nominal features as well. Most of feature selection and learning algorithms were adopted to deal with mixed feature types, but recent research shows that common machine learning algorithms benefit from treating all features in a uniform fashion (Dougherty *et al*., 1995; Hall, 1999).

Discretization is the process of transforming continuous feature values into categorical ones. The basic learning algorithms and the feature selection/ranking algorithms are developed to deal with nominal values or discrete numeric values, but also adopted to continuous numeric features. For example, CFS requires all features to be of the same type and therefore, the discretization is needed. In order to calculate Information gain the discretization of numeric features has been performed using the minimum class entropy method proposed by Fayyad and Irani (1993).

The minimum class entropy method evaluates the class entropy $H(y|f)$ using candidate partitions of a continuous feature f into two intervals as its new discrete values v_1 and v_2 in order to select a cut point T for discretization. A cut point resulting to minimum entropy is selected. Formula 17 (Section 4.2) is used for calculations of entropy of a class variable y observing the continuous feature f for each partition into two intervals, where interval μ_1 corresponds to the number of instances for which a feature f takes the value v_1, and interval μ_1 corresponds to the number of instances for which a feature f takes the value v_2.

This method is applied recursively to the two intervals of the previous partition until some stopping criterion is satisfied. The minimum class entropy method employs a stopping criterion based on the MDL principle (Rissanen, 1978).

Several studies have shown that in some cases discretization can degrade generalization accuracy (Ventura & Martinez, 1995; Ismail & Ciesielski, 2003). The method of discretization and the level of inherent error placed in the class have a major impact on classification errors generated after discretization. The general effectiveness of discretization varies significantly depending on the shape of data distribution considered. Ismail and Ciesielski (2003) have shown that highly skewed distributions or distributions having high peaks tend to result in higher classification errors, and relative superiority of supervised discretization over unsupervised discretization is diminished significantly when applied to these data distributions. However, real data do not exactly

follow the statistical distributions, thus in the case of different feature types discretization is preferable.

The scales of individual features can be drastically different due to specific measurement units use. These measurement units are not inherent characteristics of data; therefore, normalization can prevent potential problems with distance computation and feature weighting caused by different scales. In addition, features measured on the same scale might have considerably different means, variances, and possibly higher order moments, which can lead to similar problems as different scales (Fradkin, 2005). It is desirable for many feature selection, clustering, and classification techniques that features would not only be of one type (discrete numeric), but their values also would be normalized to the same norm. The values of features have to be normalized in order to ensure they are comparable and have the same effect.

If one feature has a relatively large range of values, it can overpower the other features. For example, if feature f_1 has values form 1 to 1000, and feature f_2 has only values from 1 to 10, then the influence of f_1 on distance estimation will be higher. Feature selection algorithms tend to encourage features with many values as well, as it was explained in (Skrypnyk, 2005). Therefore, feature values are often normalized by dividing by the range of that feature values. Where appropriate, the distances can be normalized instead of direct normalization of the attribute values.

It is also common to divide by the standard deviation instead of range, or to "trim" the range by removing the highest and lowest few percent (for example, 5%) of the data from consideration in defining the range. It is also possible to map any value outside this range to the minimum or maximum value to avoid normalized values outside the range [0..1]. Domain knowledge can often be used to decide which method is most appropriate (Wilson & Martinez, 1997).

The side effect of discretization is that some instances become indiscernible and therefore, duplicate instances appear in the data. Those instances have to be removed before further processing, because they will confuse the results obtained from most learning algorithms. This fact stands for discretization as a prior step to model construction.

All data sets used in the experimental study have been pre-processed in the following way. All numeric features are normalized and discretized. All nominal attributes are encoded to binary. Features with a particular dominating value appearing in no less than in 70% of instances are removed. Missing values are treated as separate values in case of nominal features, or replaced with a mean value in continuous and discrete numeric features.

The choice of an appropriate pre-processing technique is problem-specific in this study. The methodology is described in the subsections describing the experiments. Weighted inverse exponential distance function used for clustering part of BDP has two options for normalization described in Subsection 5.1.2, (1) conventional statistical normalization, where each feature independently transformed to have zero mean and unity variance, and (2)

normalization by range of feature values so that each feature takes values in the interval [0..1]. In the experiments, described in this thesis, normalization has been made according to Formula 43.

7.1.2 The imbalanced class representation problem, sample size

The imbalanced class representation problem is created by a significant difference in the number of instances representing classes in the training set. Prediction accuracy for the minority class is usually low. Often all instances are assigned to the majority class. In this case, the overall accuracy does not depict actual classifier's performance. Feature selection methods that assign scores to features considering their class (supervised feature selection) also have difficulties to produce correct estimates of feature merits. In practical applications, the ratio of the small to the large classes can be drastic, such as 1 to 100, 1 to 1000, 1 to 10000, and sometimes even more (Chawla *et al*., 2002). This is addressed in the literature as a problem of imbalanced class representation, the class imbalance problem, or skewed class distributions (Chawla *et al*., 2002, Monard & Batista, 2003).

There are several possible solutions to this problem. If the data set is large enough and the minority class includes reasonable number of instances to train a classifier, the majority class can be balanced via sampling. Thus, the first solution is to alter the training balance replicating instances from the minority class called "up-sampling" and ignoring some instances from the majority class called "down-sampling" (Weiss & Provost, 2003). Cost-sensitive learning approach provides a mechanism to control the process of learning from the minority class.

This study is limited to a standard approach to measure accuracy taking into consideration the effect of imbalanced classes (Monard & Batista, 2003). For future comparative studies one of the appropriate sampling techniques can be used.

Sample size is a critical issue in some natural domains, such as biomedicine (Mukherjee *et al*., 2003). Theoretical proof is provided, for example, in Fukunaga (1990): the required number of training samples is linearly related to the dimensionality for a linear classifier and to the square of the dimensionality for a quadratic classifier. The experiments have demonstrated that there are circumstances where second order statistics are more relevant than first order statistics in discriminating among classes in high dimensional data. In terms of nonparametric classifiers, 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, 1999)

Jimenez and Landgrebe (1999) state that it is reasonable to expect that high dimensional data contains more information in the sense of a capability to detect more classes with more accuracy. At the same time these characteristics entail that a supervised learning algorithm performing computations at full dimensionality, may not deliver this advantage unless the available labeled

data is substantial. This was proven that with a limited number of training samples there is a penalty in classification accuracy as the number of features increases beyond some point. (Jimenez & Landgrebe, 1999)

7.2 Experimental evaluation of IPA on heterogeneity variations

Relevance of features in subproblems can be evaluated using individual feature merit measures or subset merit measures. Subset merit measures output numerical estimates of different subsets and the best subset. Individual feature merit measures evaluate the contribution of each feature to discriminate between classes. Individual feature merit measures following the assumption of independence between features are called "myopic merit measures. Information gain is one of such merit measures. Individual feature merit measures that in some way take into account interactions between features while evaluating the merit of each feature are called non-myopic merit measures. (Hong, 1997) In this study, ReliefF (Kononenko, 1994) is considered as one of them.

IPA is based on comparing profiles of relevance of features in subproblems. Those profiles are ranks of features produced according to their estimated merits. Thus performance of IPA depends on the success of a feature merit measure selected. Experimental evaluation of IPA using Information gain and ReliefF feature merit measures on several data sets representing different variations of heterogeneity is considered.

7.2.1 Data sets used in the experiments

The benchmark data sets used in this study has been taken from UCI Machine Learning Repository (Blake *et al.*, 1998). Langley (1994) indicates that finding adequate data sets to test new techniques is crucial. In particular, UCI Repository has a few data sets with a substantial fraction of irrelevant features. From the previous studies on local feature relevance, for example, those described in Domingos (1997), Howe and Cardie (1997), and Apte *et al*. (1998), one may conclude that even fewer data sets from public repositories are suitable to investigate local feature relevance. The main characteristics of these data sets are summarized in Table 18.

Connect-4 Opening (CON)

This data set consists of the instances representing all legal 8-ply positions in the game of Connect-4 in which neither player has won yet, and in which the next move is not forced. From these positions it is necessary to predict either win/loss for the first player, or draw. The board contains 6 rows numbered from 1 to 6, and 7 columns marked form *a* to *g*. Each cell of the board is represented by a feature having three possible values: *x* – the first player has taken, *o* – the second player has taken, and *b* – blank. (Blake *et al.*, 1998)

The original data set contains 67557 instances with the following class

distribution: 44473 win (65.83%), 16635 loss (24.62%) and 6449 draw (9.55%). The data set used here contains 5% randomly sampled instances of the original data set with the nearly same class distribution: 2235 win (66.18%), 818 loss (24.22%) and 324 draw (9.59%).

For the Connect-4 classification task features are locally predictive. In each new unclassified instance there is no need to consider all features (values of each cell) in order to make a prediction. The features important for prediction change from instance to instance. Obviously, the subset of features taking *x* or *o* values in the particular instance is always relevant for this instance. In order to make a prediction one has to consider first the nearest cells around those that are taken. Thus for each instance the subset of features taking *x* or *o* values specifies an additional subset of features taking *b* value (the nearest cells around) to be considered. Therefore, relevance of the particular subset of *b*-valued features depends on the contextual features whose values are either *x* or *o*.

TABLE 18 Characteristics of synthetic and benchmark data sets used in the experiments. A row of the table presents the mnemonic of the data set, the number of instances included in the data set, the number of different classes of instances and the number of instances # per each class *n* (#/cl*n*), and the number of categorical and numeric features in the instances.

№	Data	# instances	# classes	# instances per class	Features	
					Categorical	Numeric
1	CON	3377	3	2235/cl1, 818/cl2, 324/cl3	42	-
2	VEH	846	4	199/cl1, 217/cl2, 218/cl3, 212/cl4	-	18
3	VOW	990	11	90/cl1, 90/cl2, 90/cl3, 90/cl4, 90/cl5, 90/cl6, 90/cl7, 90/cl8, 90/cl9, 90/cl10,90/cl11	-	10
4	VOWC	990	11	90/cl1, 90/cl2, 90/cl3, 90/cl4, 90/cl5, 90/cl6, 90/cl7, 90/cl8, 90/cl9, 90/cl10,90/cl11	2	10
5	WINE	178	3	59/cl1, 71/cl2, 48/cl3	-	12
6	1-CL HET	3000	3	1000/cl1, 1000/cl2, 1000/cl3	-	9

Vehicle Recognition Using Silhouettes (VEH)

The classification problem is to associate a given silhouette as one of four types of vehicle (double decker bus, Chevrolet van, Saab 9000, and an Opel Manta 400), using a set of features extracted from the silhouette. This particular combination of vehicles was chosen with the expectation that the bus, van and either one of the cars would be readily distinguishable, but it would be more difficult to distinguish between the cars. The vehicles have been viewed from the constrained elevation. They were rotated and their angle of orientation was measured using a radial gratitude beneath the vehicle. The data set contains 18 numerical features extracted from silhouettes of vehicles.

An investigation of rule trees indicated that the tree structure was heavily influenced by the orientation of the objects, and grouped similar object views into single decisions. (Blake *et al.*, 1998)

Vowel Recognition (VOW) with Contextual Features

Vowel data set (VOW) represents the problem of speaker independent recognition of the eleven steady state vowels of British English using a specified training set. There are 15 individual speakers, male and female, each saying each vowel 6 times. The vowels are indexed by integers 0-10. For each utterance, there are ten floating-point input values, with array indices 0-9 used as features. There are the following vowels:

vowel	word	vowel	word	vowel	word	vowel	word
i	heed	A	had	O	hod	u:	who'd
I	hid	a:	hard	C:	hoard	3:	heard
E	head	Y	hud	U	hood		

In the Vowel Context data set (VOWC) the contextual information implicit in the original data was added as two contextual features - speaker's gender and identity. The use of contextual information for this classification problem is described in Turney (1993).

Wine Recognition (WINE)

This data set has been created as the result of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. The analysis determined the quantities of 13 constituents found in each of the three types of wines having different origins. The classification task is to determine the origin of wine using results of the chemical analysis represented by 13 continuous numeric features corresponding to those 13 wine constituents. There are 3 classes corresponding to wine origins. These classes are separable. In a classification context, this is a well-posed problem with "well behaved" class structures. (Blake *et al.*, 1998)

Pure One-Class Heterogeneity synthetic data set (1-CLHET)

The Pure one-class heterogeneity data set exemplifies the case of one-class heterogeneity where a subset of features is relevant only to distinguish instances of a particular class from the other. Synthetic data is continuous for the sake of vivid visual interpretation of heterogeneity.

Local relevance of features, in this case, relevance at the instances corresponding to the particular class labeling, is represented as features taking particular non-random values for these instances. Features f_1, f_2, and f_3 follow normal distribution N(1,0) in class 1, features f_4, f_5, and f_6 follow normal distribution N(1,0) for instances of class 2, and features f_7, f_8, and f_9 follow normal distribution N(1,0) for instances of class 3 as shown in Table 19. The irrelevant features take random values following the uniform distribution U(-6,6).

TABLE 19 Synthetic pure one-class heterogeneity data set (1-CLHET).

	Features									
	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	
1	**N(0,1)**	**N(0,1)**	**N(0,1)**	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	
…	**N(0,1)**	**N(0,1)**	**N(0,1)**	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	
1000	**N(0,1)**	**N(0,1)**	**N(0,1)**	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	
1001	U(-6,6)	U(-6,6)	U(-6,6)	**N(0,1)**	**N(0,1)**	**N(0,1)**	U(-6,6)	U(-6,6)	U(-6,6)	
…	U(-6,6)	U(-6,6)	U(-6,6)	**N(0,1)**	**N(0,1)**	**N(0,1)**	U(-6,6)	U(-6,6)	U(-6,6)	
2000	U(-6,6)	U(-6,6)	U(-6,6)	**N(0,1)**	**N(0,1)**	**N(0,1)**	U(-6,6)	U(-6,6)	U(-6,6)	
2001	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	**N(0,1)**	**N(0,1)**	**N(0,1)**	
…	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	**N(0,1)**	**N(0,1)**	**N(0,1)**	
3000	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	U(-6,6)	**N(0,1)**	**N(0,1)**	**N(0,1)**	

Visual interpretation for the pure one-class data set presented in Table 19 is provided in Figure 14.

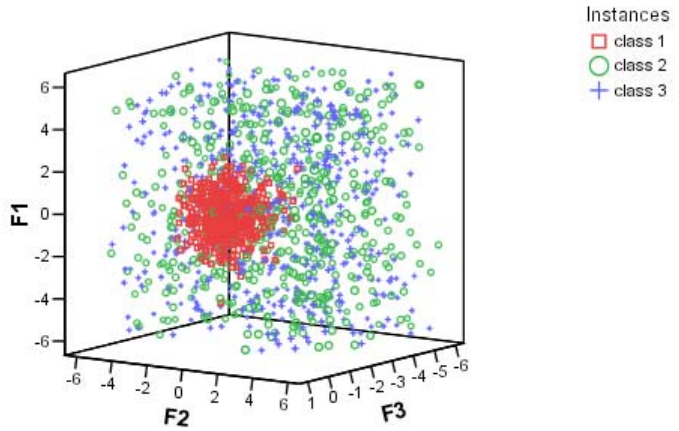

FIGURE 14 Synthetic pure one-class heterogeneity. 9-dimesional data (Table 19) presented in the subspace of features f_1, f_2, and f_3, which are relevant to distinguish instances of class 1 from the other instances. In these dimensions instances of class 1 are mainly concentrated in the interval [-1;1] with respect to features f_1, f_2, and f_3, but they are heavily interleaved with instances of classes 2 and 3 in this interval.

It shows that in the subspace of feature relevant for class 1 this class is separable from classes 2 and 3. Instances of class 2 and 3 are heavily interleaved, surrounding instances of class 1 concentrated in the center of the cloud. It is

despite the fact that in the dimensions corresponding to features f_1,f_2, and f_3, instances of class 1 are mainly concentrated in the interval [-1;1], but they are heavily interleaved with instances of classes 2 and 3 in this interval. Figure 14 shows half of the original data points for which $f_3 < 0$ in order to reveal the structure inside.

7.2.2 Experiments with feature ranking in subproblems

This subsection demonstrates how feature rankings produced by a myopic feature merit measure (Information gain) and by a non-myopic feature merit measure (ReliefF) differ in the initial problem and in the subproblems of the selected data sets after decomposition.

Table 20 presents ranks of feature merits produced using these two measures. Table 21 provided the IPA calculations based on the ranks shown in Table 20.

Considering the synthetic problem representing one-class heterogeneity one may see that ranks produced by Information gain and ReliefF are not very different. In the subproblems both measures placed locally relevant features to the top of ranks. This synthetic problem does not have interacting features, thus both measures have succeeded.

By author's preliminary results and the results obtained in Hong (1997), IPA≥ 0.3 indicates some heterogeneity and decomposition into subproblems is worthy. IPA in subproblems for 1-CLHET is about 0.4 (Table 21). For the Vehicle data set importance of features varies in subproblems as well, but the estimates produced by Information gain and ReliefF are quite different. The IPA calculations indicate that there is a significant difference in feature relevance estimates for the initial problem, where all vehicles are presented, and in the subproblem op_sb between cars. Dissimilarity between the initial problem and the subproblem bs_vn is not that significant according to IPA, because their measured parameters vary more than those of cars. Thus, the IPA values obtained for this data set can be explained in accordance to the domain knowledge available. The IPA values obtained using ReliefF are higher in general. One might assume that the numerical features extracted from the silhouette interact, and thus the ReliefF estimates are more reliable.

For the Vowel Context data set Information gain and ReliefF have produced quite similar ranks in the initial problem and the subproblems. The IPA values indicate that decomposition by the values of the "gender" feature should be successful.

Decomposition of the Connect-4 data set is not straightforward. Local feature selection for this data set might be beneficial for classification because of the simpler predictive models considering fewer features. From Table 20 one may see that ranks of features in the subproblems are very different for both Information gain and ReliefF.

TABLE 20 Feature ranking produced by Information gain and ReliefF.

Data set	Problem	Rank of features	
		Information gain	ReliefF
1-CLHET	1-clhet	4,5,8,7,9,3,2,1,6	4,7,5,8,6,9,3,1,2
	cl1_all	3,2,1,5,9,7,4,8,6	3,1,2,9,7,4,8,5,6
	cl2_all	4,5,6,3,8,7,2,9,1	5,4,6,3,8,7,2,1,9
	cl3_all	8,7,9,4,1,2,6,5,3	7,8,9,4,6,5,1,2,3
VEHICLE	vehicle	12,7,8,11,9,3,6,2,1,4,13,10,14,17,18,5,16, 15	8,18,7,12,9,3,10,11,2,17,1,13,4,6,14,15,5, 16
	op_sb	12,10,2,1,7,8,18,6,3,4,5,15,16,17,14,9,11, 13	1,15,18,6,10,5,2,3,17,13,11,4,8,12,14,7,9, 16
	op_bs	6,12,7,11,8,9,14,3,10,17,18,2,13,5,16,4, 15,1	18,3,10,14,15,2,17,11,5,6,13,16,7,9,12,8, 4,1
	op_vn	8,12,11,7,9,4,3,1,2,13,10,16,14,17,5,6,15, 18	12,9,7,8,11,10,3,2,1,4,13,17,16,18,15,6,5, 14
	sb_bs	6,12,11,7,8,9,14,3,2,13,10,18,17,5,15,1, 16,4	18,3,10,14,2,17,15,6,5,11,12,13,7,1,9,16, 8,4
	sb_vn	8,12,7,11,4,9,3,1,2,13,10,16,17,14,6,5, 18,15	12,9,7,8,11,10,2,3,1,4,13,17,18,16,6,5,14, 15
	bs_vn	12,7,8,9,11,6,1,3,4,14,13,2,5,18,17,15,10, 16	8,7,9,12,18,3,1,11,10,17,2,13,4,6,14,5,15, 16
VOWEL CONTEXT	vowc	4,3,7,6,9,11,10,8,5,12,1,2	3,4,6,7,8,12,5,11,9,10,1,2
	vowc_F	3,2,5,8,6,11,9,4,7,10,1	5,3,2,6,8,4,11,7,9,10,1
	vowc_M	2,3,6,10,7,4,11,5,8,9,1	2,3,7,6,5,10,4,11,8,9,1
CONNECT-4	connect-4	1,2,7,8,14,15,19-22,26,27,31-34,37,38	1,2,7-9,13-16,19-21,25,26,31-33,37,38
	subpr1	8,19,9,25,2,31,38,20,7,37,5,4,3,27,39,1, 21,26,22,40,33,13,32,10,34,11,6,23,35,41, 42,36,24,29,16,15,18,17,12,30,14,28	8,9,19,7,2,37,13,38,10,1,20,4,3,31,32,5,11 ,26,21,22,40,33,27,34,30,29,35,41,42,6,36 ,18,17,23,24,16,28,12,15,14,39,25
	subpr2	14,32,38,37,7,13,33,34,41,40,20,16,25,15, 2,39,8,3,31,21,9,1,35,19,10,30,6,36,42,5, 4,11,18,17,24,22,23,29,12,28,26,27	37,38,14,32,33,13,19,31,39,8,1,7,2,41,15, 40,34,16,20,4,6,5,12,10,11,17,29,28,30, 36,42,27,23,22,18,26,24,3,9,35,21,25
	subpr3	15,14,1,37,10,3,2,31,39,26,25,27,8,9,38, 19,16,33,32,17,11,13,34,7,4,28,5,6,12,36, 35,30,42,41,40,29,21,20,18,24,23,22	15,14,37,8,1,13,9,7,16,19,38,25,2,31,39,3, 26,33,32,27,10,28,4,5,12,6,35,30,29,36, 42,41,40,24,21,20,18,22,23,34,17,11
	subpr4	21,20,1,19,31,37,33,7,15,13,32,18,16,39, 22,14,17,3,9,8,25,4,2,40,23,10,34,38,24,6, 5,11,30,29,35,41,36,42,12,26,28,27	20,21,1,19,7,15,13,14,31,37,22,8,9,25,32, 2,33,18,17,16,4,38,39,11,6,5,12,29,28,27, 36,35,30,41,26,42,24,10,40,34,23,3
	subpr5	7,21,9,14,2,20,37,8,15,31,19,26,28,13,10, 22,39,1,3,38,27,25,23,29,4,16,11,5,40,42, 6,33,32,34,36,35,30,17,12,18,24,41	21,13,7,20,19,26,25,27,8,1,14,2,22,9,37,3 9,31,28,15,29,23,16,5,6,12,11,35,34,33, 41,40,36,42,17,18,24,32,30,3,10,4,38
	subpr6	2,7,1,37,26,32,31,15,27,8,33,13,9,19,25, 28,14,16,34,20,38,35,3,4,6,11,10,12,5,39, 36,30,42,41,40,29,21,18,17,24,23,22	31,32,13,7,26,33,37,1,2,14,19,8,15,27,16, 9,20,25,34,11,5,4,12,10,6,17,41,30,29,40, 39,36,22,21,18,24,42,23,3,35,28,38
	subpr7	21,37,1,25,2,23,22,31,38,13,20,32,8,7,19, 24,15,14,5,6,3,4,11,12,9,10,16,35,34,33, 36,41,40,39,30,42,18,17,26,29,28,27	21,37,1,22,2,31,25,19,20,23,8,13,7,17,16, 15,14,5,6,3,4,11,12,9,10,34,33,41,35,39, 40,36,30,26,42,18,28,29,27,24,32,38
	subpr8	15,37,26,32,7,19,2,17,38,16,1,13,18,31, 25,14,5,6,3,4,8,11,12,9,10,35,36,33,34,41, 42,39,40,30,22,23,20,21,28,29,24,27	15,1,25,38,19,16,37,2,32,17,14,26,7,18, 13,5,6,3,4,8,11,12,9,10,34,35,42,33,40,41, 36,39,30,22,23,20,21,28,29,24,27,31
	subpr9	19,14,26,37,1,20,32,38,6,16,25,23,21,3,4, 15,31,22,7,2,5,13,17,41,33,9,8,24,30,42, 28,29,10,18,39,35,36,27,12,11,40,34	19,1,14,37,13,25,5,20,7,38,6,15,16,31,32, 26,2,23,9,10,11,8,12,18,39,36,35,42,41, 40,34,28,27,24,33,30,29,21,17,22,3,4
	subpr10	37,20,7,19,13,25,21,26,11,32,9,31,10,1, 38,12,22,8,41,2,15,14,3,6,42,4,5,29,28,27, 30,35,34,33,36,18,17,16,40,24,23,39	19,37,20,7,25,13,10,31,26,11,9,1,38,32,8, 21,12,39,40,36,34,35,41,3,2,42,6,5,4,23, 29,30,27,28,24,16,15,14,17,33,18,22
	subpr11	8,37,9,14,38,13,19,7,10,29,30,3,28,2,15, 27,31,16,1,4,25,26,5,6,11,12,35,36,33,34, 41,42,39,40,20,21,17,18,24,32,22,23	37,8,7,27,28,38,13,14,19,31,1,3,9,2,25,26, 29,10,4,30,15,5,6,11,12,17,35,36,33,34, 41,42,39,40,21,20,18,22,32,24,23,16
	subpr12	2,15,20,19,1,13,16,3,35,25,4,37,7,8,34,32, 11,17,9,10,14,36,33,21,31,6,38,30,5,41, 42,39,40,23,24,18,22,26,28,29,27,12	13,1,19,7,2,15,33,32,14,20,8,34,31,3,37, 25,35,16,9,10,17,4,11,5,12,6,38,30,29,39, 42,41,40,23,24,18,22,27,28,26,36,21
	subpr13	19,1,14,8,26,21,20,31,37,15,4,3,13,16,2, 41,22,40,27,25,28,38,9,10,42,39,5,7,11,6, 12,33,32,34,36,35,30,18,17,23,29,24	19,1,37,8,13,14,26,38,25,20,15,2,21,31,3, 9,41,7,22,39,28,16,42,10,36,6,11,12,23, 32,24,29,30,18,17,34,33,35,5,4,40,27
	subpr14	1,37,15,14,26,32,4,16,28,39,25,31,2,3,5, 40,17,10,34,27,38,7,8,11,41,13,35,9,33,6, 12,29,24,30,42,36,23,19,18,20,22,21	15,14,37,26,1,3,7,25,32,8,4,31,9,2,27,28, 13,39,38,16,33,40,10,34,5,41,17,12,6,29, 24,23,42,36,30,18,19,20,22,21,11,35
	subpr15	20,7,15,22,40,39,37,4,21,32,8,31,2,1,19, 38,34,24,13,14,33,16,9,23,17,10,11,5,35, 3,18,41,6,30,29,42,26,25,36,28,27,12	20,7,31,8,19,21,15,32,37,14,38,39,13,1, 22,2,9,33,4,34,16,10,40,3,18,24,41,42,28, 6,36,30,29,26,25,12,27,17,5,11,23,35
	subpr16	14,13,20,1,19,27,26,21,8,38,15,37,31,28, 39,22,10,25,7,2,32,16,4,34,23,3,40,5,29, 33,9,18,24,17,35,36,42,30,41,12,11,6	14,13,20,38,19,26,25,1,15,37,8,27,2,21, 31,7,32,33,28,16,22,9,3,39,10,34,24,18, 41,42,6,11,30,35,12,36,4,5,40,29,23,17
	subpr17	13,32,1,14,19,20,8,26,2,38,25,37,31,7,17, 16,15,5,6,3,4,11,12,9,10,35,36,33,34,41, 42,39,40,30,22,23,18,21,28,29,24,27	13,14,26,2,1,32,8,20,19,25,31,7,37,38,17, 16,15,5,6,3,4,11,12,9,10,35,36,33,34,41, 42,39,40,30,22,23,18,21,28,29,24,27

When considering a cell (feature) taken by a player, all its neighboring cells should be considered too. It can be interpreted as a kind of interaction between features for this classification problem. By the IPA values we verify dissimilarity between the subproblems that is quite big according to the IPA values provided in Table 21. In general, IPA values provided by both Information gain and ReliefF are on the same level, 0.3-0.6.

TABLE 21 The IPA values obtained using Information gain and ReliefF between the initial problems and subproblems.

Data sets	Importance Profile Angle (IPA)	
	Information gain	ReliefF
1-CLHET	0.442058/cl1_all	0.400908/cl1_all
	0.429688/cl2_all	0.385818/cl2_all
	0.430890/cl3_all	0.389900/cl3_all
VEHICLE	0.535326/op_sb	0.926461/op_sb
	0.201226/op_bs	0.316210/op_bs
	0.200287/op_vn	0.256522/op_vn
	0.191050/sb_bs	0.329820/sb_bs
	0.192242/sb_vn	0.235890/sb_vn
	0.134379/bs_vn	0.145933/bs_vn
VOWEL CONTEXT	0.525948/vowc_F	0.334231/vowc_F
	0.534664/vowc_M	0.403693/vowc_M
CONNECT-4	0.580264/subpr1	0.595368/subpr1
	0.618577/subpr2	0.612920/subpr2
	0.595173/subpr3	0.581786/subpr3
	0.352986/subpr4	0.397846/subpr4
	0.492375/subpr5	0.519617/subpr5
	0.501064/subpr6	0.601788/subpr6
	0.491130/subpr7	0.508985/subpr7
	0.541579/subpr8	0.528261/subpr8
	0.468015/subpr9	0.517040/subpr9
	0.488271/subpr10	0.495667/subpr10
	0.611159/subpr11	0.584605/subpr11
	0.490500/subpr12	0.528872/subpr12
	0.393300/subpr13	0.381412/subpr13
	0.532395/subpr14	0.493549/subpr14
	0.454669/subpr15	0.331949/subpr15
	0.471771/subpr16	0.305274/subpr16
	0.515651/subpr17	0.489565/subpr17

7.2.3 Evaluation of classification accuracy after decomposition

From the previous section one may see that IPA has provided an indication of heterogeneity where it was expected taking into account the domain knowledge. Considering the difference in classification accuracy before and after decomposition obtained with global/local feature selection provides additional information on the advantages and disadvantages of using IPA.

For this purpose three commonly used learning algorithms, 1-Nearest Neighbor (1-NN), Naive Bayes (NB), and C4.5 decision tree (C4.5), representing different approaches to learning have been selected. Feature selection in the

initial problems and in the subproblems has been performed using CFS, and alternatively, using the wrapper feature selection. Table 8 presents feature subsets selected on an entire data set, and feature subsets selected after decomposition. Table 9 provides classification accuracy of 1-NN, C4.5, and NB.

From Table 22 one may see that in the case of pure class heterogeneity CFS and wrapper selected nearly all features, and all the basic learning algorithms reached high accuracy level (Table 23).

TABLE 22 Global vs local feature selection produced by correlation-based feature selection (CFS) and wrapper feature selection by means of three different base classifiers.

Data set	Problem	Selected features			
			Wrapper		
		CFS	1-NN	C4.5	NB
1-CLHET	1-clhet	1-9	1-9	1-6	1-9
	cl1_all	1,2,3	1-8	1-3,7-9	1-9
	cl2_all	4-6	1-9	1,4-6,8	1-9
	cl3_all	7-9	1,2,4,6-9	1,2,7-9	1-9
VEHICLE	vehicle	4-9,11,12,14-16	3,5,6,8,10-13,17,18	1,3,6,9,10-14,17,18	1,5,6,8,11,14-16,18
	op_sb	1,10,12	1,3,8,12	3,6,13,14,16-18	5,17
	op_bs	6,11,14	1,3,5,6,8-12	3-8	3,5,14,15
	op_vn	1,7,8,11,12,16	1,2,4-6,8-11,13	1,2,4-6,8-10,12,13	1,5,8,11,12,14,16
	sb_bs	3,5,6,12,14,15	2,3,5,6,8,11,13,14	3,5-7,10-12,14	2,3,5,14,15
	sb_vn	4,7-9,11,12,16	2,4-6,8-11,13	5-10,13,17	8,16
	bs_vn	4,6,7,12	3,5-9,12,14,18	3,4,6,7,12	1,4,6,8,15-18
VOWEL CONTEXT	vowc	3,4	1,3-9,11,12	1,3,4,5,10	3-11
	vowc_F	2,3,5,6,8,11	2-10	2,3,5,6	2-11
	vowc_M	2,3,6,7,10	2-10	2-5,8	2-8,10,11
CONNECT-4	con-nect-4	1-38	1,5,7,8,11,13-16,18-24,26,27,30,31,36,37,42	1,2,8,14-17,19-22,31,32	1,2,8,14,15,17,19,20,21,26,37,38
	subpr1	2,5,7,8,34,37,39,40	1,5,7,8,38	1-38	1,22,27
	subpr2	1-3,7-9,33-35,39-41	1-3,14	2,14,38	1,14,37,41
	subpr3	1-3	1,7,15	1,8,13-15,37	2,3,14,15
	subpr4	13-23,31	1,14,19,20,21,31,32	2,14,15,20-22	1,14-16,18,20,21,38,40
	subpr5	7,39	1,2,4,9,20,21,26	1-38	21,27
	subpr6	1,2	14,15,19,26,32,34	1-38	1-38
	subpr7	1,2	1,2,8,19-23,25,37	1-38	1,25,32,37
	subpr8	1,2	2,19	1-38	1,2,37
	subpr9	13-23,26	1,32	1,2,15,19,20,21,31	1-3,14,19,22,23,26,32,37,38
	subpr10	5,7,11,37,39	1-38	1-38	1-38
	subpr11	1-10	8	8	8
	subpr12	1-4,11,33-36	13,20,33,35	1-38	1,2,16
	subpr13	1	3,19-21,31	3,19,20,21,31	3,19,20,21,31
	subpr14	1,5,7,33,37,39	1,4,9,14,15,26,32,41	1,9,14,33	3,4,15,41
	subpr15	7,8,39,40	1,15,16,21,22,40	1-38	13,21,32,40
	subpr16	13-23,27	1,4,22,33	13,14,20,21	4,10,13,14,20,21
	subpr17	1,2	32	1-42	1-42

Nevertheless, in subproblems, where CFS selected the relevant features correctly, accuracy is even higher. Wrapper used more features in subproblems considering the irrelevant ones, and it's accuracy even outperformed accuracy of CFS.

For the Vehicle data set there is no prior knowledge about the difference between the subsets of features relevant in subproblems. Decomposition has promoted accuracy growth in subproblems comparatively to the initial

problem. Distinguishing between cars in the corresponding subproblem yields lower accuracy than in the other subproblems accordingly to the prior knowledge about it.

Subsets of features selected by CFS in the subproblems of Vowel Context differ in 3 features (Table 22), and using that subsets has promoted significant accuracy increase in subproblems after decomposition. Wrapper with C4.5 selected less features than wrapper with 1-NN and NB, and accuracies are nearly same for the latter two, while 1-NN is superior in accuracy. CFS is comparable to wrapper with 1-NN in accuracy.

In the majority of subproblems of the Connect-4 data set decomposition has promoted accuracy increase both with CFS and wrapper. However, in many cases, especially with wrapper, high accuracy has been obtained as a result of bad performance – almost all instances have been assigned to the majority class because of the imbalanced class representation in subproblems (*italics* in Table 23).

TABLE 23 Classification accuracy obtained on data representing the initial problems and subproblems after decomposition.

Data set		Classification accuracy, %					
		1-NN		C4.5		NB	
		CFS	Wrapper	CFS	Wrapper	CFS	Wrapper
1-CLHET	1-clhet	90.3667	90.3667	89.4333	89.3333	96.6667	96.6667
	cl1_all	90.6667	92.3000	93.2667	93.2000	94.2333	95.3333
	cl2_all	91.2333	92.8000	88.7333	93.3667	78.6667	95.2000
	cl3_all	91.6667	94.0667	93.7667	93.9000	93.7667	95.5667
VEHICLE	vehicle	64.8936	71.1584	67.6123	70.8038	49.0544	56.9740
	op_sb	54.5455	55.4779	54.7786	57.5758	53.8462	51.7483
	op_bs	89.3023	96.2791	92.3256	95.5814	72.5581	85.3488
	op_vn	84.4282	95.3771	86.6180	92.2141	82.4818	84.1849
	sb_bs	90.5747	96.5517	93.3333	92.6437	82.9885	84.8276
	sb_vn	82.9327	96.1538	84.6154	91.8269	81.7308	83.8942
	bs_vn	96.6427	98.5612	97.8417	97.8417	67.8657	85.8513
VOWEL CONTEXT	vowc	62.6263	99.0909	58.9899	81.9192	65.8586	66.9697
	vowc_F	98.0519	99.1342	78.5714	80.3030	74.6753	80.3030
	vowc_M	92.6136	97.9167	79.7348	81.0606	69.8864	78.0303
CONNECT-4	con-4	63.9621	55.6115	70.9802	61.2245	70.7137	71.7501
	subpr1	59.5469	63.4304	66.9903	68.2848	67.3139	68.2848
	subpr2	60.4167	65.2778	70.8333	71.5278	70.1389	71.5278
	subpr3	61.4458	*61.4458*	64.1566	*69.5783*	65.0602	*67.7711*
	subpr4	68.5131	61.2245	72.8863	*68.2216*	71.4286	*66.4723*
	subpr5	63.3929	*63.3929*	66.0714	*64.2857*	65.1786	*62.5000*
	subpr6	82.5243	*80.5825*	82.5243	*82.5243*	79.6117	*82.5243*
	subpr7	71.7172	65.6566	65.6566	65.6566	66.6667	*63.6364*
	subpr8	60.6061	*64.8485*	64.2424	*64.2424*	62.4242	*63.0303*
	subpr9	72.8625	*62.8253*	74.7212	*69.5167*	66.9145	*68.4015*
	subpr10	77.4436	*73.6842*	78.1955	*78.1955*	75.9398	*76.6917*
	subpr11	45.8716	*65.1376*	65.1376	*65.1376*	63.3028	*65.1376*
	subpr12	62.5806	69.6774	71.6129	71.6129	69.0323	69.0323
	subpr13	*62.0253*	65.4008	*62.0253*	68.7764	*62.0253*	66.2447
	subpr14	57.2614	61.4108	65.1452	62.2407	65.1452	63.4855
	subpr15	61.7241	65.8621	66.5517	65.5172	65.8621	63.4483
	subpr16	63.4615	66.9872	69.8718	69.5513	70.5128	68.9103
	subpr17	50.0000	*54.1667*	*75.0000*	*66.6667*	*70.8333*	*54.1667*

The experiments shown that decomposition and local feature selection on subproblems resulted in accuracy growth in the most of cases according to the expectations.

An ability of IPA to provide an indication of heterogeneity has been studied on several variations of heterogeneity: one-class heterogeneity, class

heterogeneity, feature space heterogeneity and feature space heterogeneity with contextual features.

The profile of features importance used in IPA is based on the individual contribution of every feature. The contribution of each feature is evaluated by a feature merit measure, which itself may consider a particular interaction between features (non-myopic), or may not consider feature interactions (myopic).

The experimental results with Information gain and ReliefF merit measures have confirmed that success of IPA depends on how well the selected feature ranking methods can handle presented feature interactions. Sometimes heterogeneity estimates using ReliefF and Information gain were quite different. For example, for the Vehicle data set the IPA values using ReliefF was somewhat higher, and for the Vowel Context data set, vice versa, IPA using Information gain was higher.

Indication of heterogeneity provided by IPA using feature ranking is verified via classification accuracy obtained after local feature selection in subproblems. For this purpose the non-myopic CFS feature subset merit measure based on the symmetrical uncertainty, a variant of Information gain, has been applied along with wrapper feature selection.

The experiments have shown that in the most of cases accuracy growth has been reached as a result of decomposition. These results confirm an indication of heterogeneity by high IPA values in the corresponding subproblems.

CFS is competitive to wrapper in majority of practical situations. Often wrapper yields higher accuracy, but considering different learning algorithms one may see that the subsets of features selected in subproblems are very different. Wrapper's performance depends on inductive bias of a learning algorithm.

The most important issues related to the heterogeneity problem are building a set of models and combining those models for the final prediction. Finding the effective way to combine models is an important task of future research.

7.3 Experiments with cancer survival prediction data

Cancer survival analysis is commonly used to assess cancer treatment programs and to monitor the progress of regional and national cancer control programs (Green *et al.*, 2002). It is also used in medical research to answer the questions such as: how do particular circumstances or treatments increase or decrease the odds of survival and what is the fraction of a population which will survive past a certain time? Of those that survive, at what rate will they die or fail? Can multiple causes of death or failure be taken into account? How do particular circumstances or characteristics increase or decrease the odds of survival?

Cancer survival rates are defined by the percentage of people who survive a certain type of cancer for a specific amount of time. It helps to put survival statistics in perspective establishing whether a certain cancer type is relatively easy or more difficult to cure, and what are the factors that influence this statistics. Researchers have developed tools called nomograms that can be used to predict cancer outcomes or assess risk based on specific characteristics of a patient and of his or her disease. These tools are pioneered by researchers at Memorial Sloan-Kettering Center, NY, USA, to help patients and physicians make important treatment decisions (Delen *et al*., 2005).

Cancer survival prediction is an example of applications, where high predictive accuracy is a secondary goal, while determining and understanding the risk factors is primary. Considering the large number of diverse factors, either presented in the data or hidden, one can assume existence of subclasses or survival groups, where the outcome is influenced by different factors, relatively common within each group. Bidirectional data partitioning is applied in order to discover data structure. Further data analysis reveals relative importance of factors that influence the outcome that has local relevance of the descriptive features, speaking in data mining terms.

7.3.1 SEER cancer data description

The SEER Cancer data sets are provided by the Surveillance, Epidemiology and End Results (SEER) Program of the National Cancer Institute (SEER, 2011), the authoritative source of information about cancer incidence and survival in USA. Data used in the experiments are combined from SEER 9, 13, and 17 Registries Databases for patients diagnosed in the period 1973-2008, released in April 2011. SEER 9 registries are Atlanta, Connecticut, Detroit, Hawaii, Iowa, New Mexico, San Francisco-Oakland, Seattle-Puget Sound, and Utah. In this data set, cases diagnosed from 1973 through 2008 are available for all registries except Seattle-Puget Sound (1974+) and Atlanta (1975+). The database contains one record for each of 4,021,996 tumors. Cases are associated with the population data using three racial groups: White, Black, and Other. The Other race category used in the SEER 9 Registry database consists of American Indian/Alaska Native and Asian/Pacific Islander combined. All descriptions related to data fields and data format provided in this thesis refer to this data version. Population and mortality data collections, provided along with the primary data collection, are not used in the survival time analysis. Published reports on survival time prediction using data mining techniques are based on previous versions of SEER data (Delen *et al*., 2005; Fradkin *et al*., 2006; Sarvestani *et al*., 2010; Bellaachia & Guven, 2006).

The primary data collection includes separate records on 9 site-specific cancer groups: (1) breast cancer, (2) colon and rectum cancer, (3) digestive system cancers, (4) female genital cancer, (5) male genital cancer, (6) lymphoma of all sites and leukemia, (7) respiratory apparatus cancer, (8) urinary apparatus cancer, and (9) all other sites cancers.

Descriptive features contain two types of information: demographical and clinical. Demographical information includes age, sex, race/ethnicity, place of birth, etc. Clinical information includes history of previous diseases and treatments, diagnostic information, such as location of the disease, its type (morphology, histology) and extent, the types of treatment (radiation, surgery) and cause of death, where applicable. Original data cases (instances) are stored in rows containing fields (features) of fixed length (124 features in 276 positions). The database version used in this thesis is available at http://www.seer.cancer.gov/.

7.3.2 SEER data pre-processing

SEER database has evolved over time and therefore certain information is only available in recent years. For our analysis goal, we have selected only records collected in the period 1998-2008, as in 1998 several particularly significant features were introduced.

Original data sets include cases diagnosed in 1973-2008. During the period of data collection treatments has changed, new categories (features) were introduced and data collected from new locations was added (in SEER 11 and 13 databases). All features are listed in Table 24.

We used a combined data set based on SEER 9, 13, and 17 databases. Duplicate instances corresponding to multiple entries of the same patient have been removed from the data set.

For our analysis goal, we have selected only records collected in the period 1998-2006 from SEER 9 database, as in 1998 several particularly significant features were introduced.

Survival time recode variable (#112) is originally calculated using the data of diagnosis and one of the following: date of death, date last known to be alive, or follow-up cutoff date used for a current version of data, December 31, 2008.

Based on previous studies of SEER Cancer data sets in the data mining community (Delen *et al*., 2005; Fradkin *et al*., 2006), the analysis goal is to predict 8 months survival time. 8 months is a median survival time has been established as a cut-off for short-time survival (class 1) and long-term survival (class 0). A class variable is obtained as follows:

- if survival time is unknown, or a patient died within 8 months, but not from a specific cancer type, or a patient was diagnosed in 2008, alive by December 31, 2008, and survival time is less than 8 months such instances will be discarded from consideration;
- if a patient died within 8 months and the cause of death is a specific cancer, the instances are assigned to class 1;
- if a patient's survival time is more than 8 months, the instances are assigned to class 0;
- all remaining instances, where class cannot be determined, are discarded.

Cause of death can be determined incorrectly as a metastatic site (SEER: Measures of Cancer Survival, 2011). This processing step discards these cases from consideration.

TABLE 24 SEER feature names in the original encoding. Years are provided if features were not available during the entire period of SEER data collection. Information on applicable years, where not specified, can be obtained from data. More details are can be found at http://seer.cancer.gov/data/documentation.html.

№	Name	Years	№	Name	Years
1	Patient ID number	-	63	RX Summ-Reconstruct 1st	1998-2002
2	Registry ID	-	64	Reason for no surgery	-
3	Marital Status at DX	-	65	RX Summ-Radiation	-
4	Race/Ethnicity	-	66	RX Summ-Rad to CNS	1988-1997
5	Spanish/Hispanic Origin	-	67	RX Summ-Surg / Rad Seq	-
6	NHIA Derived Hispanic Origin	-	68	RX Summ-Surgery Type	1973-1997
6	Sex	-	69	RX Summ-Surg Site 98-02	1998-2002
8	Age at diagnosis	-	70	RX Summ-Scope Reg 98-02	1998-2002
9	Year of Birth	-	71	RX Summ-Surg Oth 98-02	1998-2002
10	Birth Place	-	72	SEER Record Number	-
11	Sequence Number--Central	-	73	Over-ride age/site/morph	-
12	Month of diagnosis	-	74	Over-ride seqno/dxconf	-
13	Year of diagnosis	-	75	Over-ride site/lat/seqno	-
14	Primary Site	-	76	Over-ride surg/dxconf	-
15	Laterality	-	77	Over-ride site/type	-
16	Histology (92-00) ICD-O-2	-	78	Over-ride histology	-
17	Behavior (92-00) ICD-O-2	-	79	Over-ride report source	-
18	Histologic Type ICD-O-3	-	80	Over-ride ill-define site	-
19	Behavior Code ICD-O-3	-	81	Over-ride Leuk, Lymph	-
20	Grade	-	82	Over-ride site/behavior	-
21	Diagnostic Confirmation	-	83	Over-ride site/eod/dx dt	-
22	Type of Reporting Source	-	84	Over-ride site/lat/eod	-
23	EOD-Tumor Size	1988-2003	85	Over-ride site/lat/morph	-
24	EOD-Extension	1988-2003	86	SEER Type of Follow-up	-
25	EOD-Extension Prost Path	1985-2003	87	Age Recode <1 Year olds	-
26	EOD-Lymph Node Involv	1988-2003	88	Site Recode	-
27	Regional Nodes Positive	1988+	89	Site Rec with Kaposi and Mesothelioma	-
28	Regional Nodes Examined	1988+	90	Recode ICD-O-2 to 9	-
29	EOD-Old 13 Digit	1973-1982	91	Recode ICD-O-2 to 10	-
30	EOD-Old 2 Digit	1973-1982	92	ICCC site recode ICD-O-2	-
31	EOD-Old 4 Digit	1983-1987	93	SEER modified ICCC site recode ICD-O-2	-

(continues)

SEER Cancer data include many variables, where numeric code is used to describe a categorical/nominal variable. For example, feature V99 - histology recode for brain groupings that has 29 code values. Two of those values correspond to all other brain histologies, not included in previous 27 values, and non-brain recode. Some values are associated with a certain cancer type. For breast cancer data V99 has a constant value.

TABLE 24 (continues)

№	Name	Years	№	Name	Years
32	Coding System for EOD	1973-2003	94	ICCC site recode ICD-O-3	-
33	Tumor Marker 1	1990-2003	95	ICCC site recode extended ICD-O-3	-
34	Tumor Marker 2	1990-2003	96	Behavior Recode for Analysis	-
35	Tumor Marker 3	1998-2003	97	ICD-O Coding Scheme	-
36	CS Tumor Size	2004+	98	Histology Recode-Broad Groupings	-
37	CS Extension	2004+	99	Histology Recode-Brain Groupings	-
38	CS Lymph Nodes	2004+	100	CS Schema v0202	-
39	CS Mets at Dx	2004+	101	Race recode (White, Black, Other)	-
40	CS Site-Specific Factor 1	2004+	102	Race recode (W, B, AI, API)	-
41	CS Site-Specific Factor 2	2004+	103	Origin recode NHIA (Hispanic, Non-Hisp)	-
42	CS Site-Specific Factor 3	2004+	104	SEER historic stage A	-
43	CS Site-Specific Factor 4	2004+	105	AJCC stage 3rd edition (1988-2003)	-
44	CS Site-Specific Factor 5	2004+	106	SEER modified AJCC Stage 3rd ed (1988-2003)	-
45	CS Site-Specific Factor 6	2004+	107	SEER Summary Stage 1977 (1995-2000)	1995-2000
46	CS Site-Specific Factor 25	2004+	108	SEER Summary Stage 2000 (2001-2003)	2001-2003
47	Derived AJCC T	2004+	109	Number of primaries	-
48	Derived AJCC N	2004+	110	First malignant primary indicator	-
49	Derived AJCC M	2004+	111	State-county recode	-
50	Derived AJCC Stage Group	2004+	112	Survival time recode	-
51	Derived SS1977	2004+	113	Cause of Death to SEER site recode	-
52	Derived SS2000	2004+	114	COD to site rec KM	-
53	Derived AJCC-Flag	2004+	115	Vital Status recode	-
54	Derived SS1977-Flag	2004+	116	IHS Link	-
55	Derived SS2000-Flag	2004+	117	Summary stage 2000 (1998+)	1998+
56	CS Version Input Original	2004+	118	AYA site recode	-
57	CS Version Derived	2004+	119	Lymphoma subtype recode	-
58	CS Version Input Current	2004+	120	SEER Cause-Specific Death Classification	-
59	RX Summ-Surg Prim Site	1998+	121	SEER Other Cause of Death Classification	-
60	RX Summ-Scope Reg LN Sur	2003+	122	CS Tumor Size/Ext Eval	2004+
61	RX Summ-Surg Oth Reg/Dis	2003+	123	CS Lymph Nodes Eval	2004+
62	RX Summ-Reg LN Examined	1998-2002	124	CS Mets Eval	2004+

Coding systems used in cancer research have been revised several times since 1973 and new coding systems have been introduced. SEER Cancer data in some cases include a separate variable for a newly introduced code, and in some cases old codes are converted to a new system, while this conversion is stored in a separate variable. These data characteristics are particularly representative in terms of classification heterogeneity.

Such codes often include contextual information. For example, feature V90 – recode ICD-O-2 to 9.

Data includes four racial groups: White, Black, American Indian/Alaska Native, and Asian/Pacific Islander. In SEER 9 database they are encoded as White, Black, and Other. It also includes the ethnic groups Hispanic and Non-Hispanic, which are not mutually exclusive from White, Black, American Indian/Alaska Native, and Asian or Pacific Islander. Therefore, these data require decomposition for predictive models construction.

Similar to other analyses described at SEER, we encode variable *Year of birth* dividing all patients onto 19 age groups (< 1, 0-4, 5-9,..., 85+).

7.3.3 Experiments with respiratory apparatus cancer data

The respiratory apparatus cancer data set used in the experiments has been extracted from the respiratory data. The pre-processed version of the data set was granted by the DIMACS researchers that investigated it with application of *k*-means clustering (Fradkin, 2006), SVM and logistic regression (Fradkin *et al*., 2005).

The data set used in the experiments contains no missing values. All categorical and ordinal features were encoded as binary features. Patient survival time used as a class variable is encoded as short-term and long-term with a cut-off value of 8 months. There are 45 features, including age, race, gender, diagnosis age, region of birth, cancer site, grade, medical treatments, histology, etc. Data includes 120,318 training instances (66,923 and 53,395 in two classes) and 97,240 test instances (52,849 and 44,391 in two classes). The larger class includes patients with more than 8 months survival time, the negative class.

Specificity depicts the proportion of patients who passed 8 month survival time which are correctly identified. Sensitivity shows percentage of patients that did not pass 8 month survival time who are correctly identified.

The baseline result is obtained with 10-folds CVM performed in one run. J48 pruned decision tree has 67.07% accuracy, 69.7% specificity, and 63.9% sensitivity[2]. F-measure as a weighted average over two classes is 67.1%.

BDP with 5 iterations of weight adaptation, DBSCAN (estimated ε = 0.005), minPoints = 21, IPA threshold = 0.3, clustering inside classes (CIC), obtains 2 groups. The accuracy with J48 without feature selection is 70.18%, specificity is 71.5%, sensitivity is 68.6%, and weighted average F-measure is 70.20%. Our experiments with BDP identified that having DBSCAN assigned all instances of the training set to the same group except for some instances qualified as noise should deliver the original performance of the base classifier that does not use weights. However, instances discarded as noise (about 3%) lead to decrease in accuracy for about 4%. Therefore, we conclude that some loss

[2] Sensitivity and specificity are the most widely used statistics used to describe a diagnostic test. Sensitivity is a probability of a positive test among patients with disease specificity is a probability of a negative test among patients without disease.

in performance of BDP is credited to non-optimal settings for parameters in a clustering algorithm. Increasing ε may reduce noise, but decreasing ε will not help finding larger number of smaller subgroups (as they probably do not exist according to DBSCAN), it will just increase noise.

As *k*-Means is used on the entire data set with 8 clusters as a parameter, only 5 clusters were found (WEKA's implementation allows that). With IPA threshold = 0.25 these subgroups are not merged. Accuracy is 70.06%, specificity is 71.00%, sensitivity is 68.9%, and weighted average F-measure is 70.10%. If only 5 clusters are specified as a parameter, the clustering process goes differently, 5 clusters are found, and performance is slightly lower.

Having 4 clusters specified on the entire data set resulted in 4 subgroups found, two of which are merged with IPA < 0.25. However, the results are slightly better with 3 final pure-class groups – accuracy is 70.62%, specificity is 71.7%, sensitivity is 69.3%, and weighted averaged F-measure is 70.07%. With 3 clusters the accuracy is 70.51%, specificity is 73.10%, sensitivity is 67.4%, weighted average F-measure is 70.05%.

Manipulating the number of clusters and IPA threshold in BDP with *k*-Means in this manner we tried to find the number of clusters that corresponds to better accuracy of weighted *k*-NN (pure-class groups).

Clustering inside classes did not produce drastic changes in accuracy, more clusters were found: *k*-Means with CIC, 4 clusters per class, has found 8 subgroups. After using them as final pure-class groups with IPA threshold = 0.25, *k*-Means accuracy is 70.01%, specificity is 72.4%, sensitivity is 67.10%, and weighted average F-measure is 70.00%. With pure-class subgroups functionality of BDP corresponds to a weighted *k*-Nearest Neighbor that takes advantage of distances transformed by weights. Having the same groups by *k*-Means, we have compared performance of IPA-based agglomerative merging (in this case, weighted *k*-NN) to performance of a meta-classifier that builds a J48 decision tree over group (cluster) labels, this way predicting which base classifier to select. The performance of *k*-Means with meta-classifier has been lower: accuracy is 67.68%, specificity is 69.5%, sensitivity is 65.5%, and weighted average F-measure is 67.7%.

Manipulating the number of clusters and IPA threshold in BDP with *k*-Means we tried to find the number of clusters that corresponds to better accuracy of weighted *k*-NN (pure-class groups).

Furthermore, we have explored pairwise and one-against-all combinations of subgroups in a meta-classifier with the same 5 clusters obtained by *k*-Means on the entire data set, which is another way to merge subgroups. Only in this case, instead of merging according to feature weight profiles or class separability, we can explore all possible combinations. This approach can provide additional insights of data structure, such as hierarchical structure (one-against-all subgroups). In pairwise combinations, if both include only instances of one class, that combination will not have an important vote in the combination scheme, but meaningful combinations of different subclasses might have.

Using the same settings as above, but instead of giving group labels to multi-class meta-classifier, we used WEKA's Multi-Class Classifier with J48 with pairwise and one-against-all combinations. The results obtained with pairwise combination are the following: 70.07% accuracy, 71.00% specificity, 68.90% sensitivity, and 70.1% weighted average F-measure.

The most important features appeared in groups are related to surgery recommended or performed, radiation, and histology code. The homogeneous regions are exhibited in the age of diagnosis and region of birth features. Some features appeared in subproblems have specific extension code, which is hard to interpret by a non-expert.

7.3.4 Summary on cancer survivor analysis

SEER Cancer data sets exemplify the problem of unstable feature relevance in real-world practical tasks. These data sets demonstrate to which extent the problem of unstable feature relevance may appear in real-world practical tasks. In these data sets, unstable feature relevance appears to some extent.

Regional and time period factors appear to be important in determining data structure. This may be due to different environmental factors such as air and water pollution, nuclear and chemical accidents in the area, industrial waste discharges, etc. On a larger scale, even preferences in lifestyle could be related to a regional factor. For example, Los Angeles is well-known for popularization of sports, healthy habits, body weight control - all known to be risk-lowering factors of cancer and other diseases. These are example of hypotheses provided to epidemiology experts for further testing by application of data partitioning technique. Data partitioning results may suggest a need for collection of measurements for additional features, or extracted latent features.

7.4 Prediction of cancer types using microarrays

The recent development of high-throughput genomics technologies has enabled researchers to take a comprehensive and high-resolution view of the genetic and epigenetic changes present in cancer cells. A new field of cancer genomics studies abnormalities that promote cancer development. These include changes in DNA sequence and organization, DNA copy number, gene and microRNA expression, alternative splicing, DNA methylation, and histone modifications. Due to rapid accumulation of genomics data its successful translation into meaningful clinical end points has proven difficult. Complex, high-dimensional cancer genomics data face a challenge of integration, modeling, and knowledge discovery. (Cancer Genomics Workshop Materials, 2011)

Microarray chips enable measuring the expression level (i.e. the intensity of the expression) of thousands of genes simultaneously under different conditions or in different tissues. Therefore, specifics of classification problems in bioinformatics include redundant representation of the problem in a high

dimensional feature space where each feature corresponds to a gene, and the number of instances, each corresponding to a single experiment on measuring gene expression, is fairly limited. Typically, an instance is presented by a number of genes that are irrelevant, insignificant, or redundant to the classification problem at hand (Zhou & Mao, 2005).

Data mining techniques are successfully applied in gene expression analysis. In particular, functional relationships among genes are discovered finding groups of instances, where certain subsets of genes have similar expression behavior. Knowledge on functional relationships among genes is important for understanding gene regulation. Application of standard clustering methods is limited for this kind of task. Seeking subsets of genes as features with respect to groups of instances as microarray chip experiments is imposed by the existence of a number of experimental conditions where the activity of genes is uncorrelated (Madeira & Oliveira, 2004). Therefore, this domain is a target application for Bidirectional Data Partitioning in classification problems, having subspace clustering as an established tool for clustering tasks.

Bioinformatics and biomedicine are rapidly changing fields with explosive growth of advanced technologies. Many new directions have been explored in the last decade. The problems described below are used for illustration purposes and do not necessarily reflect current state of research in these fields. Genetics basics needed for understanding the background of the problem domain are provided in the Appendix 5.

7.4.1 Predicting the type of cancer based on gene expression data

Cancer is a genetic malady, mostly resulting from acquired mutations and epigenetic changes that influence gene expression. Accordingly, a major focus in cancer research is identifying genetic markers that can be used for precise diagnosis or therapy. (Tamayo *et al*., 2002) Having collection of tumor tissue samples from patients with cancer, researchers study the genome of these cancer samples for clues about cancer's root cause, how it thrives and spreads, and how to stop it. The Broad Institute of Harvard and MIT, MA, USA, has several dedicated platforms to collect and track information on cancer tissues prior to analysis. MIT Center for Genome Research, Whitehead Institute, Cambridge, MA, USA, has released data sets containing gene expression data open to public, which can be accessed from ELVIRA Biomedical Data Set Repository (ELVIRA, 2011), or in a raw format (Cancer Program Data Sets, 2011).

Microarrays are used to get clues about which genes are expressed to control cell, tissue or organ function. By measuring the level of RNA production for every gene at the same time, researchers can learn the genetic programming that makes cell types different and diseased cells different from healthy ones. Different types of microarray use different technologies for measuring messenger RNA expression levels. Affymetrix arrays are oligonucleotide microarrays, which are currently the most popular commercial arrays

(Pitetsky-Shapiro & Tamayo, 2003). Affymetrix GeneChip platform generate absolute expression values from a single sample

Biological samples (tissue samples) are homogeneous or heterogeneous mixtures of different cell types (e.g. malignant cells with varying degrees of differentiation, stromal elements, blood vessels, and inflammatory cells). Two tumors with similar clinical stages can vary markedly in grade and in relative proportions of different elements (e.g., prostatic adenocarcinoma). Tumors of different grades might potentially differ in gene expression, and different markers can be expressed either by malignant cells or by other cellular elements.

Because heterogeneity in the sample population can complicate the interpretation of gene expression studies, sample selection is an important issue that must be kept in mind when analyzing gene expression data. (Tamayo *et al*., 2002)

Two major sources of variation include biological variation and variation due to technical factors. The latter has been significantly reduced with technologic improvements. To understand the range of biologic variation the number of samples is increased. However, the process of obtaining additional biological samples is often expensive, involved, and time consuming (Mukherjee *et al*., 2003). Therefore, high-level analysis techniques in data mining should compensate for heterogeneity in data affecting performance of predictive modeling.

Leukemia (AML-ALL) data set.

The leukemia data set is described in Golub *et al*. (1999). This data set contains gene-expression levels measured from bone marrow and peripheral blood samples of 72 patients with either acute lymphoblastic leukemia (ALL, 47 instances) or acute myeloid leukemia (AML, 25 instances) for 7129 human genes (features). ALL and AML are two classes. All the samples were derived from patients at the time of diagnosis before chemotherapy. The raw data are available among Cancer Datasets from Broad Institute (Cancer Program Data Sets, 2011) and at ELVIRA Biomedical Data Set Repository (2011). The original raw data is generated by the Affymetrix Gene Chip microarray scanning software. The task is to predict one of two cancer types based on a subset of genes relevant to discriminate between ALL and AML. The goal is to see if the type of cancer can be predicted based solely on gene expression monitoring. Authors in Golub *et al*. (1999) pointed out that two classes were detected by clustering without prior knowledge of classes. Therefore, importance of this study extends to development of a general strategy for discovering and predicting cancer classes. This also suggests that application of BDP in discovering subclass/superclass data structure has a potential in this kind of problems.

The original data generated by Affymetrix's GeneChip software includes markers for gene presence in a sample. Thus, each sample has two pieces of data associated with it: an expression value for a gene and an Absent/Marginal/Present (A/M/P) call. The A/P calls are an indication of the

confidence in the measured expression value. These markers are currently ignored in GeneCluster2, software produced by the research team of Whitehead Cancer Genomics Group, and by many other researches who presented results on this and other related data sets (Zhou & Mao, 2005; Wang *et al*., 2007; Ramaswamy *et al*., 2001).

Our preliminary results obtained after partitioning the entire data set with BDP have discovered that class ALL is densely grouped in feature f_{2280} (gene M84526) and f_{1179} (gene M19507), while class AML is densely grouped in feature f_{758} (gene D88270) and feature f_{4680} (gene X82240). Figures 15 and 16 demonstrate 2-dimensional projections of this data set. However, this clear result is explained by markers for genes absence. Genes D88270 and X82240 are absent in most cases of AML and present in most cases of ALL. Genes M19507 and M84526 are vice versa, absent in most cases of ALL and present in most cases of AML. There are other genes that expressed a similar pattern. This result connects with the cancer prediction results reported by Wang *et al*. (2007). They have quantified gene expression levels which can be considered as supervised discretization. This quantification is meant to represent gene regulation information that is implicitly represented in gene expression data. This information is also reflected in gene expression markers.

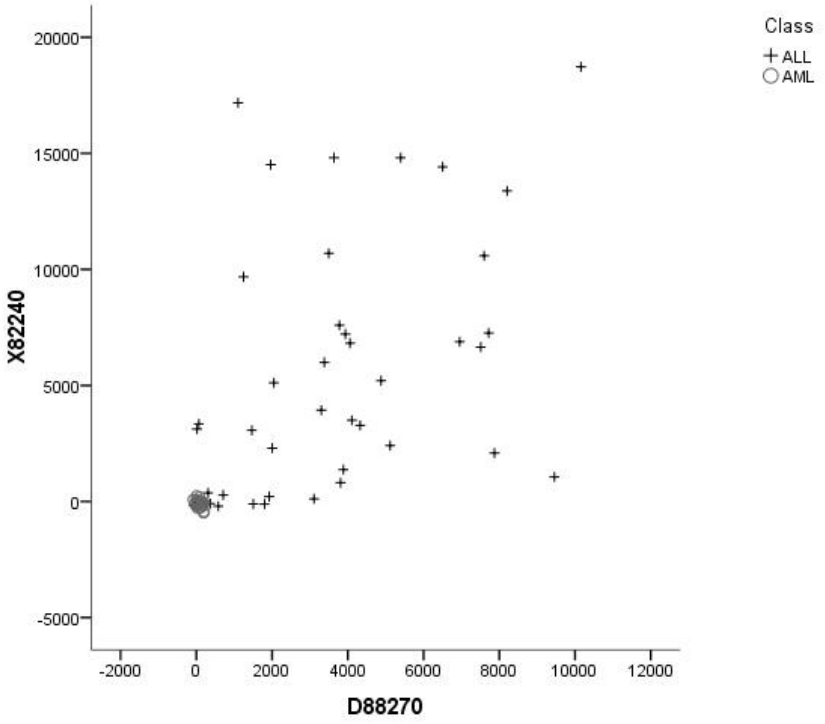

FIGURE 15 ALL_AML Leukemia data set, genes D88270 and X82240. 72 samples from bone marrow and peripheral blood as instances, expression levels for 7129 genes as features. Shown in two dimensions corresponding to genes D88270 (GB DEF = (lambda) DNA for immunoglobin light chain) and X82240 (TCL1 gene (T cell leukemia) extracted from H.sapiens mRNA for Tcell leukemia/lymphoma 1).

According to Wang *et al*. (2007), from a biological viewpoint, gene expression tends to be controlled by gene regulation activities. It appears that biological phenotypes are more directly correlated with gene regulation than gene expression results. For this reason, gene regulation information can be extracted from microarray data for cancer classification to avoid the harmful effects of noise and errors at the gene expression level. As a result, a set of genes have been selected following a global filter feature selection approach.

The results presented below are performed using the original set of genes without normalization. Normalization (standardization) is applied if one is interested in emphasizing relative rather than absolute differences in gene intensity and therefore, optional (Tamayo *et al*., 2002).

Class discovery by means of clustering performed in Golub *et al*. (1999) aimed to identify fundamental subtypes of cancer in general, and finer subclasses of Leukemia, in particular. Using SOM in a modification of the GeneCluster computer package (Reich *et al*., 2004), they have obtained the following 4 clusters. Immunophenotype data on the instances within clusters have shown that four clusters largely corresponded to AML, T-lineage ALL, B-lineage ALL, and B-lineage ALL, respectively (Golub *et al*., 1999). One subgroup is exclusively AML, another contains all 8 T-ALLs, and last two subgroups contain the majority of B-ALL instances. The results suggested that the latter two subgroups might best be merged into a single class.

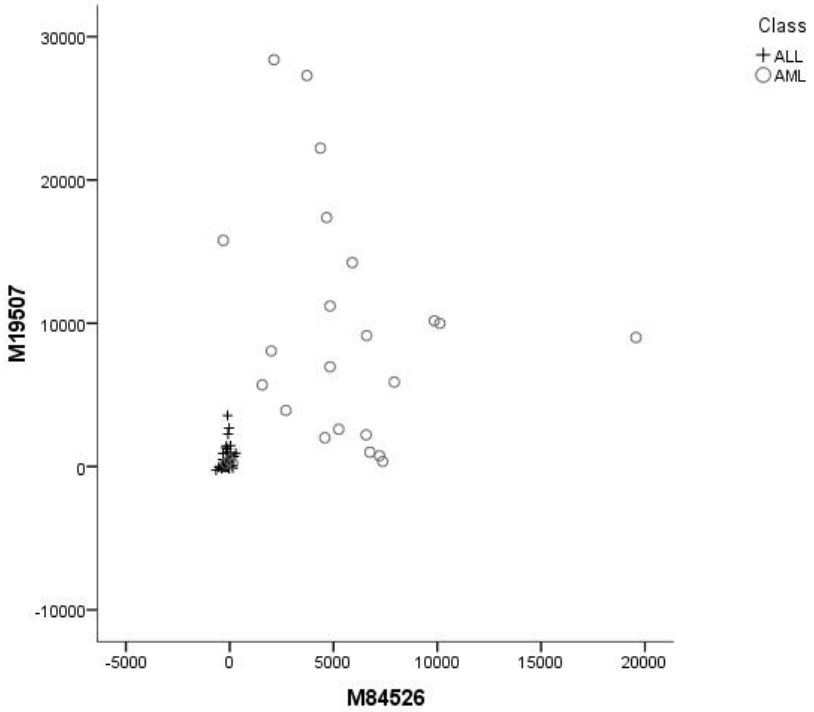

FIGURE 16 ALL_AML Leukemia data set, genes M19507 and M84562. 72 samples from bone marrow and peripheral blood as instances, expression levels for 7129 genes as features. Shown in two dimensions corresponding to genes M19507 (MPO Myeloperoxidase) and M84562 (DF D component of complement (adipsin)).

In our experiments, clustering has been performed on the entire data set. 4 clusters have been discovered with an average-linkage distance-based hierarchical clustering with 3 instances as outliers.

Clusters as subgroups and classes are placed in the following correspondence. Instances indices (first instance labeled "1"):

Subgroup 1: 28 29 30 31 32 33 35 36 38 60 61 62 65 67 68 70 71 72 (AML).
Subgroup 2: 1, 3, 4, 6, 7, 8, 9, 23, 27, 40, 44 (AML); 37 (ALL).
Subgroup 3: 12, 21, 45, 46, 47, 48, 49, 50, 56, 58 (ALL); 34, 59, 63, 64, 66, 69 (AML).
Subgroup 4: 2, 5, 10, 11, 13, 14, 15, 16, 18, 19, 20, 22, 24, 25, 26, 41, 42, 43, 51, 52, 53, 54, 55 (ALL).
Noise: 17, 39, 57 (ALL).

12 genes from the rank top are selected in each group to compare with 12 most important genes identified on the entire data set in (Zhou & Mao, 2005).

Gene importance profiles based on weights are shown in Figure 17. Profiles are shown for all genes that appear as top 12 in one or more subgroups.

4 genes with the highest ranks in subgroups:

1: D88270, L33930, X82240, M11722.
2: U46499, M84526, M19507, D88422.
3: Y09616, X95190, M23323, M12886.
4: M84526, U46499, M27891, D88422.

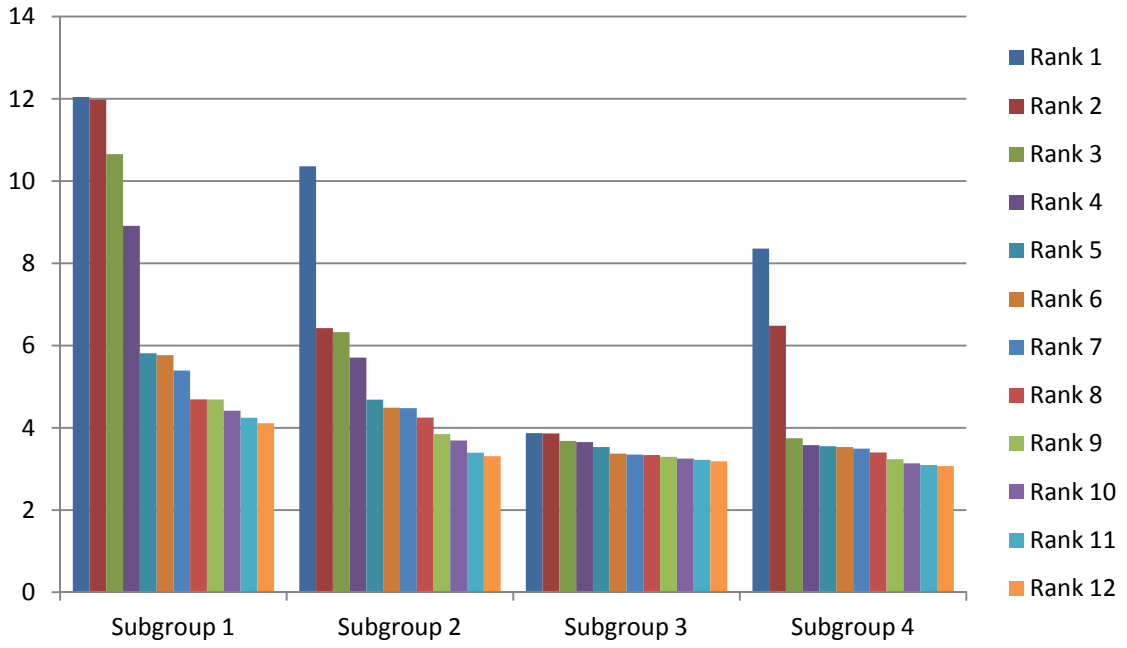

FIGURE 17 Gene importance profiles in 4 subgroup discovered in ALL_AML Leukemia data set.

BDP with weight adaptation performed in 5 iterations over 8 nearest neighbors, $\lambda = 0.2$, $\beta = 2$, DBSCAN with -radius = 0.1023 and minPoints = 4, clustering on entire data set, has detected only 2 pure-class subgroups. With IPA lower

than 0.2885 they were not merged back to the original data set, hence distance-based classifier combination scheme possessed functionality of weighted *k*-NN. Using meta-classifier resulted in functionality of the base classifier with weight information lost. Accuracy for weighted *k*-NN and J48 is 86.3014% and 87.6712 correspondingly.

Preliminary experiments have shown that DBSCAN was not able to find subclasses. It has assigned 43 instances to one pure-class subgroup and 23 to another. 6 instances are filtered as noise. CFS raised accuracy of J48 to 87.5%.

BDP with weight adaptation performed in 5 iterations over 8 nearest neighbors, *k*-Means with 4 clusters per class, clustering inside classes, has detected 8 subgroups on entire data set. With IPA agglomerative merging at IPA = 0.3, subgroups were merged into 2 groups, where the smallest subgroup included 14 instances, 3 ALL and 11 AML. The accuracy obtained with such partitioning using J48 is 90.2778%. Lowering IPA to 0.225 resulted in 4 groups and 84.7222% accuracy. Using the original 8 groups with meta-classifier resulted in 91.6667% accuracy.

The success of data partitioning cannot be considered with respect to accuracy only. The same result on weighted distances used in three different clustering algorithms with their parameters settings leaded to different results, not to mention different ensemble integration schemes and possibly different base classifiers. For example, J48 has embedded feature selection. Therefore, feature selection performed by means of feature weighting had no effect on accuracy.

As for manual selection of 4 features discovered using bidirectional data partitioning, the improvement in accuracy is considerable: J48 – 91.6667%, BDP with *k*–Means, clustering inside classes, 4 clusters per class, meta-classifier combination scheme with J48 – 93.0556%, BDP with *k*–Means, clustering inside classes, 4 clusters per class, IPA threshold = 0.2 with J48 (3 final groups) – 94.4444%, BDP with *k*–Means, clustering inside classes, 4 clusters per class, IPA threshold = 0.02 with Multi-Class Classifier and J48 – 95.8333%.

Cancer class prediction represents an important paradigm in molecular classification. The simplest analysis involves selecting the features (genes) most correlated with a phenotypic distinction of interest. These features or "marker genes" are biologically interesting in themselves, but they can also be used as the input of a classification algorithm that uses existing instances with known class labels (samples) to build a model to predict class labels for future samples. For example, marker genes (a subset of relevant features) in a cancer data set are given as an input to a classifier to distinguish cancer types on the basis of site / cell of origin or clinical outcome. (Tamayo *et al*., 2002) This data set has been extensively studied in the literature, in particular using filter feature selection. Published report on gene selection by gene using ranking techniques for Leukemia data includes 100 top genes obtained using 8 gene ranking techniques (Su *et al*., 2003).

7.4.2 Summary on cancer types discovery

According to medical research in cancer diagnostics, nowadays there is no general approach to identify new cancer types (classes) or for assigning tumors to known categories (classes). Cancer classification relies on clinical information, such as data related to patient age, gender, race, history of previous diseases and treatments, diagnostic information, etc. together with histopathological information of diseased tissue obtained after examination of tissues with surgery, biopsy or autopsy. The interpretation of both information types is subjective and tends to place tumors in currently known cancer classes based on the tissue of origin of the tumor.

Clinical information often can be incomplete or misleading. In addition, there is a wide spectrum of cancer morphology and many tumors are atypical or lack of morphological features. These difficulties can result in diagnostic confusion and hinder patient care, add expenses and confront the results of clinical trials. (Ramaswamy *et al*., 2001)

Recent studies have demonstrated feasibility of cancer classification based solely on gene expression monitoring by DNA microarrays (Golub *et al*., 1999). This generic approach suggests a general strategy for discovering and predicting cancer classes for other types of cancer, independent of previous biological knowledge.

The accurate classification of human cancer based on anatomic site of origin is an important component of modern cancer treatment. In many cases cancer type can be difficult to classify using standard clinical and histopathological approaches. Molecular approaches to cancer classification have the potential to effectively address these difficulties. However, decades of research in molecular oncology have yielded few useful tumor-specific molecular markers. An important goal in cancer research, therefore, continues to be the identification of tumor specific genetic markers and the use of these markers for molecular cancer classification.

Oligonucleotide microarray-based gene expression profiling allows investigators to study the simultaneous expression of thousands of genes in biological systems. In principle, tumor gene expression profiles can serve as a molecular fingerprints that allow for the accurate and objective classification of tumors. A medical research group at the Whitehead Institute for Biomedical Research, USA, has developed computational approaches with application of supervised and unsupervised learning using gene expression data to accurately distinguish between two common blood cancer cases (acute lymphocytic and acute myelogenous leukemia) (Golub *et al*., 1999). The classification of primary solid tumors, by contrast, is a harder problem due to limitations with sample availability, identification, acquisition, integrity, and preparation. Moreover, a solid tumor is a heterogeneous cellular mix and gene expression profiles might reflect contributions from non-malignant components further confounding classification. In addition, there are intrinsic computational complexities in making multi-class classification.

8 SUMMARY AND CONCLUSIONS

This chapter summarizes the contributions made and the conclusions gained from analytical and experimental investigations of classification heterogeneity. Classification heterogeneity and its variations have been extensively studied and a decomposition approach under ensemble framework has been proposed. Decomposition is generally a synergy of class encoding and local feature selection. Local predictive models are constructed for each homogeneous region of a data set and applied to new instances in combination or selectively. The search strategy for finding candidate homogeneous regions is suggested for each heterogeneity type. The future perspectives for development and extension of this approach are outlined at the end.

8.1 Summary

Decomposition of heterogeneous classification problems is performed in this thesis within an ensemble framework. It is proposed to perform such decomposition via selection of instances representing homogeneous regions.

Classification heterogeneity phenomenon has been described and variations of the basic heterogeneity types have been considered. Heterogeneous classification problem can be characterized as having the class boundaries that are very different in different regions of the feature space. So, importance and contribution of different features to discriminate between classes varies across the set of instances.

Heterogeneous data has some structure related to grouping of instances at homogeneous regions and class labeling. This structure is usually unknown prior to learning. Therefore, data characteristics revealing this structure can be estimated using information-theoretic, statistical, and geometrical descriptors. It is suggested that local feature interactions and local predictive ability of individual features are important characteristics related to heterogeneity that can be estimated using feature selection heuristics.

Class heterogeneity type is prevalent in certain practical tasks because of specifics of the processes generating the data used for prediction. As a result, data sets have a structure related to class labeling, and heterogeneity is exhibited around classes. Data structure in the case of classification heterogeneity differs in homogeneous regions having different relevant features and being represented by instances from different classes, or subsets of classes.

Practical tasks often exemplify so called class heterogeneity, where grouping of instances in homogeneous regions is related to class labeling. In the case of class heterogeneity covering the representative instances is more straightforward, since homogeneous regions correspond to particular classes, or subsets of classes. For this case decomposition based on local feature-feature and feature-class interactions is suggested.

In order to approximate grouping of instances in the case of class heterogeneity class encoding has been applied. The ensemble technique combining class decomposition and local feature selection is developed for this type of heterogeneity. A decomposition scheme for ensemble generation has been described. Decomposition within an ensemble learning framework makes use of the accuracy improvement mechanisms provided by ensemble learning. In particular, the use of several learning models generated to cover one homogeneous region has promoted increase of prediction accuracy.

Three feature merit measures employed in feature selection methods have been experimentally studied regarding their ability to estimate local feature-feature and feature-class interactions. A correlation-based feature subset selection based on the Symmetrical Uncertainty measure have been used in combination with one-per-class and pairwise class decomposition schemes in the experiments on the natural and synthetic data sets.

During the experimental study various aspects of using this combined ensemble technique have been studied. They include specifics of the basic learning algorithms, representation of the training set (feature types, class representations, irrelevant and redundant features), and integration of the component classifiers in ensemble.

The Importance Profile Angle (IPA) measure is considered as providing an additional indication of heterogeneity. In the experimental study IPA has been calculated for the initial problem and subproblems obtained after decomposition. Class separability and complexity measures applied to classification problem at hand may point out that heterogeneity is not present.

The results of the analytical and empirical study described in this thesis are encouraging and may be considered as an important step towards developing specific solutions for practical problems.

8.2 Conclusions

Over the years researchers in data mining, machine learning, and other related fields have come up with various algorithms and improvements to existing approaches. Currently, there is a vast majority of supervised learning algorithms that have proven success for different applications. Some techniques and approaches are considered as most popular in data mining and contribute to trend setting (Wu *et al*., 2008). But on a larger scale, no one method is "best" because its applicability depends on the data characteristics.

In practice, every algorithm takes advantage of certain characteristics and relationships of a given data set. If those characteristics and relationships vary across the data set one deals with heterogeneity in a classification problem. The idea of bidirectional data partitioning is driven by this problem of heterogeneity, which was recognized more than ten years ago and continuously evolved since then. The goal of bidirectional partitioning is to automatically detect heterogeneity uncovering data structure, model subproblems, and combine those local models using the strengths of well-known divide-and-conquer approach.

The goal was accomplished via a synergy of feature selection/weighting, classification, and clustering considering them in a single framework and appealing to their common basic criteria of class separability and at the same time employing the power of a traditional ensemble approach.

The following conclusions have been drawn from the empirical investigation of applicability of the combined ensemble technique derived from the ideas of classification heterogeneity decomposition.

The experimental study has confirmed known factors affecting performance of the feature ranking and selection methods used to perform local feature selection, and the basic learning algorithms used to implement the component classifiers. Among those factors are imbalanced class representations, globally irrelevant, redundant and interacting features, combination of different feature types, small sample size providing insufficient representation of the classification problem, and intrinsic complexity of a classification problem whenever the analysis goal and the data collection goal did not fully coincide. Understanding these factors is crucial in interpretation of the obtained results, especially with such multi-component technique as BDP.

Different feature selection and classification methods fall under influence of those factors to a different extent. Besides, each learning algorithm has its own bias in building a learning model as well as each method evaluating merit of an individual feature or a feature subset has its own bias in producing estimates.

Relevance of the features to discriminate between classes can be determined evaluating a subset of features, or each feature individually. Evaluation of individual features independently of other features may skip important feature interactions, for example, when two features have a discriminative power together, but each of them contributes to class discrimination independently. However, some feature merit measures take into account feature interactions indirectly, for example, ReliefF considered in this thesis. Evaluation of a feature subset is more effective, but also is a more expensive method because of time required to evaluate feature subsets. As an example of such a method, correlation-based feature subset selection has been used in the study. This feature selection method can be comparative or superior to the methods evaluating features individually in many cases. However, correlation-based feature subset selection cannot reveal higher order feature interactions, as contextual features sometimes have. In this thesis it is shown that correlation-based feature subset selection can be successfully applied in decomposition of class heterogeneity, where grouping of instances in subproblems is not defined by higher order dependencies. A possible approach based on pairwise combination of features to produce one feature or binary encoding of feature values is described.

The experimental results have revealed that retaining both correlated features, retaining one of them, or discarding one of them is appropriate in different situations depending on the existing interaction between features. Hence, different individual feature / feature subset selection methods succeed in different situations being based on different assumptions about feature interactions.

The conclusion made from the experimental investigation of one-per-class decomposition is that the major problem is an imbalanced representation of two-class subproblems, especially when the initial classification problem contains many classes. Pairwise class decomposition helped to overcome this problem in many cases. In addition, with pairwise class decomposition more component classifiers are generated, thus homogeneous regions are better approximated and accuracy of ensemble prediction is higher in general than with one-per-class decomposition. Nevertheless, for the data sets with many classes pairwise class decompositions would be impractical. In general, the 1-Nearest Neighbor learning algorithm is more resistant to imbalanced class representation compared to Naïve Bayes, and especially C4.5.

As the goal of the experimental study was to discover specific domains where the approach is applicable, the comparative studies with many different data sets for statistically significant results were not performed. After experimental investigation of class heterogeneity decomposition, the conclusions are drawn taking into account the magnitude of standard deviation. Changes in accuracy are considered as significant when standard deviations are relatively small. When standard deviations are of the same order of magnitude as accuracy improvement, these results are not significant. Experiments with BDP are mostly illustrative.

There are alternative approaches to handle imbalanced class representations. Stratified sampling is a typical solution, but it can be applied with one-per-class decomposition only if the minority class in a subproblem is represented by a sufficient number of instances. Another possible approach is to apply cost-sensitive learning, since different errors are not equally destructive.

In BDP the problem of imbalanced class representation transforms to a problem of imbalanced group representation. The following regulation mechanism is applied: a designated parameter is used to control the minority group capacity. Minority group is merged with another group during IPA-based agglomerative merging. Noise group, which should be kept to a reasonable minimum with clustering parameters, is not participating in model construction.

Application of correlation-based feature subset selection (CFS) using Symmetrical Uncertainty with one-class and pairwise class decompositions has been justified in many cases. As a result, features individually predictive of the class are used in the feature subspace projections for homogeneous regions. However, in some situations heterogeneity is exhibited via the presence of contextual features, as for example in the Vowel context data set. Relevance of primary features depends on the values of contextual features, i.e. higher order dependencies take place. Correlation-based feature subset selection assuming feature independence given the class does not perform well in this case. The Information gain and ReliefF feature merit measures are applied to produce ranks of features. Those ranks can be used in feature selection, for example, within the correlation based approach, or in order to gain additional information about the presence of heterogeneity by computing the Importance Profile Angle (IPA) between feature relevance vectors.

When decomposition is performed for the data set without any prior knowledge about heterogeneity, IPA calculations between subproblems may indicate the presence of heterogeneity. Then feature merits obtained using Information gain or ReliefF for the initial problem and subproblems are used in feature merit vectors between which IPA is measured. Success of their application depends on the properties of the particular data set, mainly, on feature-feature and feature-class interactions.

Because some of the factors described above are always present in the natural data, from the pool of selected natural data sets only a few came under influence of the accuracy improvement mechanisms provided by a combined class decomposition ensemble with local feature selection. For several data sets the effects produced by class decomposition and feature selection separately have diminished in combination. However, in several cases accuracy has been increased only in combination of class encoding and local feature selection.

The lack of consensus in the results produced by different feature ranking methods and instability of their individual results has been a subject of research in data mining community for quite some time (Boz, 2002; Alelyani *et al*., 2011; Wang & Khoshgoftaar, 2011; Gao *et al*., 2011). Therefore, either selection of a

pre-defined number of features or selection by a cut-off value can greatly affect the accuracy. There are methods to overcome this problem including parameter tuning, but for the sake of simplicity and, in some cases, feasibility, these approaches are not applied here acknowledging the fact that current performance is not the best possible.

Integration scheme for the two-class component classifiers considered in this study greatly depends on how good the component classifiers are in determining probability of being in one of the classes. In order to get a correct classification with this integration scheme probability that an unclassified instance belongs to class B of *D*-1 classifiers should be greater than 50%, but smaller than probability that this instance belongs to class A produced by the right classifier to use, where *D* is the number of classes.

Calculations of the Importance Profile Angle is helpful in detection of heterogeneity despite the fact that it is heavily affected by imperfections of the obtained feature merits. In the cases accuracy increase has been reached after decomposition, IPA may indicate whether it is due to the presence of heterogeneity in the initial problem, or some other reasons. When IPA is applied as a measure of similarity between feature weight profiles, reaching consensus is not an issue.

Often in practical tasks the prior knowledge about the data structure related to heterogeneity is fairly limited or not available. In such cases the homogeneous regions to be modeled as subproblems can be estimated, for example, using some information theoretic, geometrical or statistical descriptors (Ho & Basu, 2002). The study is aimed to cover applicability of BDP, and the entire classification heterogeneity decomposition approach, describing additional class separability and complexity measures as well as indirect heterogeneity testing procedures, which at least can point to non-existence of heterogeneity in the domain of interest and can be useful in preliminary data analysis.

High computational complexity of BDP, mostly due to distance-based clustering and weight adaptation. is well justified by a natural setup, when investment to computational resources is saving time of a human expert that looks for decomposition of a complex classification problem formulated on voluminous high-dimensional data onto simpler subproblems.

Currently, the most promising results of classification heterogeneity decomposition approach application are obtained in biomedicine, in particular, in molecular classification of cancer types. The primary goal of cancer genomics research with application of predictive data mining is discovery of cancer subclasses and local gene interactions. BDP has a great potential in this domain being conceptually based on local feature relevance discovery and decomposition. This technique is very flexible and can be easily extended to target problems associated with every particular project or study in cancer genomics. Future application domains are described in Section 8.4.

8.3 Limitations and future work

The study of classification heterogeneity phenomenon described in this thesis has opened many other research directions regarding decomposition of heterogeneous classification problems.

The combined ensemble technique for heterogeneity decomposition can be extended for all variations of heterogeneity since the way to partition the set of instances for decomposition is determined. The alternative decomposition schemes may be developed for different variations of classification heterogeneity. The key to developing such decomposition schemes is in the investigation of data structure and characteristics related to different types and variations of heterogeneity. In addition, predictive performance is greatly influenced by the basic learning algorithm and an ensemble integration scheme selected.

Comparative analysis of learning algorithms that relate their performance to data characteristics has received attention only recently. Typically the measurements applied to data are some statistical or information theoretic descriptions. In classification tasks, where instances are assigned with class labels, it is also important to measure geometrical characteristics of class distribution. (Ho, 2002)

Some measures may highlight the manner in which classes are separated or interleaved. Therefore, another approach to heterogeneity decomposition is based on geometrical data characteristics (class boundaries, and structure of clusters regarding class labeling) that serve to find subsets of instances representing homogeneous regions simultaneously analyzing different feature subspace projections.

Investigation of different types and variations of classification heterogeneity will be continued aiming to develop a solution for a general case of classification heterogeneity using geometrical approach to approximate the homogeneous regions in the data. Applicability of spectral clustering to find homogeneous regions is currently under research.

Additional theoretical and experimental work has to be performed before the ideas of decomposition using an ensemble learning framework for heterogeneous classification problems can be routinely used in practice. The study resulted in this thesis has the following limitations that should be taken into account in further research on this topic.

Geometrically, the decomposition approach applied expands only to discriminating by hyperplanes orthogonal to the axes of features in the feature space where instances reside. The problem of finding new or derived features other than given features can be addressed in the future. A liner combination or other functional combination of several features may contribute to separating out homogeneous regions.

The problems with data that require pre-processing, for example, noisy data, missing feature values, small data sample, imbalanced class

representation, and so on, may significantly influence predictive performance and smooth the effect of application of the technique proposed, as has been demonstrated in the experimental study. Such data problems should be regarded as a separate issue along with the problem of mixed feature types. A number of experimental trials and data sets used in experiments can be increased for obtaining statistically sound conclusions.

In this thesis decomposition approaches are developed for class heterogeneity, and a general case, feature space heterogeneity. Approach based on contextual features is described, but not implemented. It remains a topic of future research considering some preliminary results obtained in this thesis and in Apte *et al*. (1998). Class heterogeneity decomposition is limited to one-per-class and pairwise class encoding. Whenever one-per-class encoding is applied to a general class heterogeneity case, the feature subsets used in different component classifiers will be interleaved. Pairwise encoding method may generate too many models in some situations. Other class encoding methods will be evaluated in further studies.

Another research direction to be explored to eliminate this limitation is related to increasing the number of local models covering homogeneous regions to make use of the mechanisms of accuracy improvement of ensemble learning. Accuracy of the ensemble constructed on locally relevant features for class heterogeneity can be further improved using direct error minimization approach as in traditional sampling techniques, such as boosting and bagging. By the preliminary results considered in Skrypnyk *et al.* (2003), incorporation of boosting increases a number of learning models providing better coverage of homogeneous regions.

Integration of local learning models to predict class membership of new unclassified instances is a crucial issue in development of the decomposition approach based on the ensemble learning framework. Alternative integration schemes based on selection will be considered in the future. Promotion of local regions coverage optimization by means of stochastic discrimination theory is another potential way to improve the existing technique.

Addition of randomizing component would enhance ensemble based on ensemble theory. That is during approximation of local regions one may introduce boosting or random clustering and create partitioning multiple times to cover for errors. Such approaches have been widely explored before and added to increased performance. Applicability of other class separability measures and clustering techniques may provide improvements in certain situations, or in general. Information-theoretic based criteria for class separability and clustering are currently under investigation.

As every method in data mining, BDP has its advantages and disadvantages, situations where it will produce good results and situations where it will fail. BDP includes many components: feature weighting, distance function, clustering, feature selection, ensemble integration, and classification with a base classifier. All of those components influence performance of BDP. Their standalone performance is extensively studied in the literature. For

example, weight adaptation is implemented using *k*-Nearest Neighbor technique and Manhattan distance function. *k*-Nearest Neighbor is not suitable for data with varying density. In high dimensions, data is sparse and the concept of similarity may not be meaningful anymore. *k*-Nearest Neighbor is computationally expensive, but using Manhattan distance instead of Euclidean helps to lower computational costs. Distance-based clustering techniques also depend on a distance function choice. Reaching consensus between different clustering techniques is a known problem in data mining. Clustering techniques in BDP produce very different results; therefore, particular choice should depend on data specifics. DBSCAN, having many advantages, is also sensitive to varying densities and high dimensions. Uneven density can be attributed, in particular, to a hierarchical data structure. BDP targets this problem with weight regulation in a distance function. Still, if different classes have different densities, DBSCAN requites different parameter settings in the clustering-inside-classes mode. Otherwise, most instances are considered as noise. Estimation of the radius parameter inside classes leads to additional computational expenses and not always effective. *k*-Means tends to produce equal-sized clusters, which is not suitable for situations where subgroups are very unequal. One way to overcome this problem is to increase the number of clusters, and then subsequent application of IPA-based agglomerative merging procedure will join clusters that belong to one subgroup.

Empirical evaluation of BDP has uncovered many specifics not associated with the use of particular component techniques. IPA-based agglomerative merging procedure can merge two subgroups at a time when sometimes it is preferable to merge all candidate subgroups in one go. The latter has it is own drawbacks, therefore agglomerative merging was the choice. Experimental evaluation has shown that in many situations, even a less than ideal merging from domain knowledge point of view still leads to improved classification performance.

Multiple components of BDP have dual influence on the final result. On one side, their choice is complicated without proper understanding of how those components work and what their parameters mean overall leading to a poor result which cannot produce any meaningful knowledge about the problem domain or show increased performance either. On the other side, consensus of the components can lead to consolidation covering for each other's drawbacks. That is a main reasoning for designing most of the combined techniques in data mining.

YHTEENVETO (FINNISH SUMMARY)

Skrypnyk, Iryna
Epävakaiden ominaisuuksien merkitys luokittelutehtävissä
Jyväskylä: Jyväskylän yliopisto, 2011
Väitöskirja

Viime vuosikymmenen aikana tapahtunut tiedonkeruuteknologioiden kehittyminen on muuttanut tiedonlouhinnan luonnetta huomattavasti. Tänä päivänä yksi tiedonlouhinnan haasteista on rakenteeltaan yhä monimutkaisemmaksi käyvän datan käsittely. Tämän seurauksena datan sisältämissä muuttujissa/piirteissä esiintyy usein epävakaisuutta. Toisin sanoen, merkityksellisten piirteiden joukko ei ole sama läpi koko havaintoaineiston. Tarkastellessa tätä ongelmaa toisesta näkökulmasta, data sisältää lokaaleja aliavaruuksia, joissa relevanttien piirteiden joukot eroavat toisistaan. Globaalit mallit eivät täten kykene tuomaan oleellista tietoa esille datarakenteista.

Tässä väitöskirjassa kuvataan epävakaiden piirteiden relevanssiongelma luokittelutehtävien yhteydessä. Työssä käsitellään yksityiskohtaisesti myös heterogeenisten luokitteluongelmien käsitettä sekä erityyppisten piirreavaruuksien heterogeenisuutta. Väitöskirjassa esitellään myös osatehtävistä johdettu monimalliratkaisu. Osatehtävissä annettu luokittelutehtävä on jaettu ryhmittelemällä joukoksi yksinkertaisempia, paremmin separoituvia tehtäviä, joista kullekin voidaan luoda oma malli. Ratkaisu esitetään kokonaisuusmallin viitekehyksessä. Piirteiden relevanssin osalta epävakaiden luokitteluongelmien hajotelmien muodostamiseen ehdotetut hakustrategiat perustuvat luokkien suhteen erilaisiin tarkkuustasoihin. Nämä kandidaattiosatehtävät arvioidaan piirteiden relevanssiprofiilien kautta. Profiilit esitetään painovektoreina, jotka saadaan piirteiden hyötyä kuvaavista mitoista tai vaihtoehtoisesti etäisyysmittojen sovittamisen tuloksena. Muita kompleksisuusmittoja, kuten luokkien erottelevuutta ja tiheyspohjaisia mittareita, ehdotetaan hajotelmien arvioimiseen ja alustaviin heterogeenisuustesteihin.

Tämä tutkimus edistää luokittelutehtäviin liittyvien data-analyysitavoitteiden saavuttamista ja valottaa tiedon rakenteisiin sekä monimutkaisuuteen liittyviä näkemyksiä. Vaikutuksia luokittelukykyyn tutkittiin lukuisten biolääketieteen tutkimusalueelta saatujen synteettisten ja aitojen aineistojen sekä yleisesti käytettyjen testiaineistojen testaamisen kautta. Tässä tutkielmassa havaittiin, että osatehtävien louhinta on useissa tapauksissa mahdollista ja se tuottaa merkityksellisiä tuloksia datan osittamisessa. Monissa tapauksissa se osoitti johtavan myös menetelmien parempaan ennustuskykyyn.

REFERENCES

Aggarval, C.C., Hinneburg, A. & Keim, D.A. 2001. On the surprising behavior of distance metrics in high dimensional space. LNCS, Vol. 1973, London, UK: Springer, 420-434.

Aha, D., Kibler, D. & Albert, M. 1991. Instance-based learning algorithms. Machine Learning 6(1), 37-66.

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

Alelyani, S., Liu, H. & Wang, L. 2011. The effect of the characteristics of the dataset on the selection stability. In Proc. of Twenty Third Int. Conf. on Tools with Artificial Intelligence. Los Alamitos, CA: IEEE CS Press, 970-977.

Ali, K. & Pazzani, M. 1996. Error reduction through learning multiple descriptions. Machine Learning 24(3), 173-202.

Allen, J.D. 1990. Expert play in Connect-Four. Available at [http://homepages.cwi.nl/~tromp/c4.html] (refereed: December 16, 2011).

Allen, M.P. 1997. Understanding regression analysis. New York, NY: Springer-Verlag.

Anand, R., Methrotra, K., Mohan, C.K. & Ranka, S. 1995. Efficient classification for multiclass problems using modular neural networks. IEEE Trans. on Neural Networks 6(1), 117-125.

Apte, C., Hong, S.J., Hosking, J., Lepre, J., Pednault, E., & Rosen, B. 1998. Decomposition of heterogeneous classification problems. Intelligent Data Analysis 2(1-4), 81-96.

Batchelor, B.G. 1978. Pattern recognition: ideas in practice. New York: Plenum Press.

Bauer, E. & Kohavi, R. 1999. An empirical comparison of voting classification algorithms: bagging, boosting, and variants. Machine Learning 36(1-2), 105-139.

Bellaachia, A. & Guven E. 2006. Predicting breast cancer survivability using data mining techniques. In Kamath, C. & Burl, M. (Eds.) Proc. of Ninth Workshop on Scientific Data Mining in conjunction with the Sixth SIAM Int. Conf. on Data Mining. Philadelphia: SIAM Press, 3-7.

Bernadó-Mansilla, E. & Ho, T.K. 2005. Domain of competence of XCS classifier system in complexity measurement space. IEEE Trans. on Evolutionary Computation 9(1), 82-104.

Biberman, Y. 1994. A context similarity measure. In F. Bergadano & L.D. Raedt (Eds.) Proc. of European Conference on Machine Learning, LNCS, Vol. 784. Berlin Heidelberg: Springer-Verlag, 49-63.

Blake, C., Keogh, E. & Merz, C. 1998. UCI Repository of machine learning databases. Available at [http://archive.ics.uci.edu/ml/] (refereed: December 16, 2011).

Blayo, F., Cheneval, Y., Guérin-Dugué, A., Chentouf, R., Aviles-Cruz, C., Madrenas, J., Moreno, M. & Voz, J.L. 1995. Enhanced learning for evolutive neural architecture. ELENA, R3-B4-P: Benchmarks. ESPRIT Basic Research Project 6891, Spain. Technical report.

Blum, A. & Langley, P. 1997. Selection of relevant features and examples in machine learning. Artificial Intelligence 97(1-2), 245-271.

Bontempi G. & Birattari M. 2005. From linearization to lazy learning: A survey of divide-and-conquer techniques for nonlinear control. Int. Journal of Computational Cognition 3(1), 56-73.

Boz, O. 2002. Feature subset selection by using sorted feature relevance. In Wani, A.M., Arabnia, H.R., Cios, K.J., Hafeez, K. & Kendall, G. (Eds.) Proc. of First Int. Conf. on Machine Learning and Applications. Los Alamitos, CA: IEEE CS Press, 147-153.

Breiman, L. 1996. Bagging predictors. Machine Learning 24(2), 123-140.

Breiman, L. 2000. Randomizing outputs to increase prediction accuracy. Machine learning 40(3), 229-242.

Breiman, L., Friedman. J., Olshen, R. & Stone, C. 1984. Classification and regression trees. New York, NY: Chapman & Hall.

Cardie, C., & Howe, N. 1997. Improving minority class prediction using case-specific feature weights. In D. Fisher (Ed.) Proc. of Fourteenth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 57-65.

Cancer and Cancer Genetics. 2011. Available at [http://www.stonybrook.edu/ovprpub/cmm/cancergenetics.html] (refered: December 16, 2011.)

Cancer Genomics Workshop Materials. 2011. Mathematical and Computational Approaches in High Throughput Genomics, Institute of Pure and Applied Mathematics, University of California at Los Angeles. Available at http://www.ipam.ucla.edu/programs/genmini (refereed: December 5, 2011).

Cancer Genome. 2011. Available at [http://cancergenome.nih.gov/cancergenomics/whatisgenomics/whatis] (refereed: December 16, 2011).

Cantú-Paz, E. 2004. Feature subset selection, class separability, and genetic algorithms. In Proc. of Sixth Int. Conf. on Genetic and Evolutionary Computation. Berlin Heidelberg: Springer-Verlag, 959-970.

Chawla, N., Bowyer, K.W., Hall, L.O. & Kegelmeyer, P. 2002. Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research 16(1), 321-357.

Cover, T.M. & Thomas, J.A. 1991. Information Theory. New York, NY: Wiley & Sons.

Cunningham, P & Carney, J. 2000. Diversity versus quality in classification ensembles based on feature selection. In R. Lopez de Mantaras & E. Plaza (Eds.) Machine Learning: Proc. of Eleventh European Conference on Machine Learning, LNCS 1810. Berlin: Springer, 109-116.

Dash, M. & Liu, H. 1997. Feature selection for classification. Intelligent Data

Analysis 1(1-4), 131-156.
Davis, J. & Goadrich, M. 2006. The relationship between precision-recall and ROC curves. In W.W. Cohen & A. Moore (Eds.) Proc. of Twenty Third Int. Conf. on Machine Learning. New York: ACM Press, 233-240.
Delen, D., Walker, G. & Kadam, A. 2005. Predicting breast cancer survivability: a comparison of three data mining methods. Artificial Intelligence in Medicine 34(2), 113-127.
Devijver, R.A. & Kittler, J. 1982. Pattern recognation: A statistical approach. London: Prentice Hall.
Dey, D., Sarkar, S. & De, P. 2002. A distance-based approach to entity reconciliation in heterogeneous databases. IEEE Trans. on Knowledge and Data Engineering 14(3), 567-582.
Diday, E. 1974. Recent progress in distance and similarity measures in pattern recognition. In C.J.D.M. Verhagen (Ed.) Proc. of Second Int. Joint Conference on Pattern Recognition, 534-539.
Dietterich, T. 1997. Machine learning research: four current directions. Artificial Intelligence Magazine 18(4), 97-136.
Dietterich, T. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 10(1), 1985-1923.
Dietterich, T. & Bakiri, G. 1995. Solving multiclass learning problems via error-correcting output codes. Artificial Intelligence Research 2(1), 263-286.
Domencioni, C., Papadopoulos, D., Gunopulos, D. & Ma, S. 2004. Subspace clustering of high dimensional data. In Proc. of Fourth SIAM Int. Conf. on Data Mining. Philadelphia, PA: SIAM Press, 517-521.
Domingos, P. 1995. Rule induction and instance-based learning: A unified approach. In Proc. of Fourteenth Int. Joint Conference on Artificial Intelligence, Vol. 2. San Mateo, CA: Morgan Kaufmann, 1226-1232.
Domingos, P. 1997. Context-sensitive feature selection for lazy learners. Artificial Intelligence Review 11(1-5), 227-253.
Domingos, P. 2002. Machine learning. In W. Klosgen & J. Zytkow (Eds.) Handbook of Data Mining and Knowledge Discovery. New York: Oxford University Press, 660-670.
Domingos, P. & Pazzani, M. 1997. On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning 29(2), 103-130.
Dougherty, D., Kohavi, R. & Sahami, M. 1995. Supervised and unsupervised discretization of continuous features. In A. Prieditis & S.J. Russell (Eds.) Proc. of Twelfth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 194-202.
Duda, R. & Hart, P. 1973. Pattern classification and scene analysis. New York, NY: John Wiley & Sons.
Duda, R.O., Hart, P.E. & Stork, D.G. 2001. Pattern classification. New York, NY: John Wiley & Sons.
Dümbgen, L. & Zerial, P. 2011. On low-dimensional projections of high-dimensional distributions. Publications in Statistics Theory, Cornell University Library, eprint arXiv:1107.0417.

Efron, B. & Tibshirani, R. 1993. An introduction to the bootstrap. New York: Chapman & Hall.

Everitt, B.S., Landau, S. Leese, M., & Stahl, D. 2011. Cluster analysis. London: Wiley.

Fayyad, U.M. & Irani, K.B. 1993. Multi-interval discretization of continuous-valued attributes for classification learning. In R. Bajcsy (Ed.) Proc. of Thirteenth Int. Joint Conference on Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 1022-1027.

Fayyad, U., Piatetsky-Shapiro, G. & Smyth, P. 1996. From data mining to knowledge discovery: an overview. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth & R. Uthurusamy (Eds.) Advances in Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI/MIT Press, 1-36.

Fisher, R. 1936. The use of multiple measurements in taxonomic problems. Annals of Eugenics 7(1), 179-188.

Fradkin, D., Dona Schneider, D. & Muchnik, I. 2005. Machine learning methods in the analysis of lung cancer survival data. Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, NJ. Technical Report 2005-35.

Fradkin, D. 2006. Within-class and unsupervised clustering improve accuracy and extract local structure for supervised classification, Department of Computer Science, Rutgers University, New Brunswick, NJ, USA. Ph.D. Thesis.

Freund, Y. & Schapire, R. 1996. Experiments with a new boosting algorithm. In L. Saitta (Ed.) Proc. of Thirteenth Int. Conf. on Machine Learning. San Francisco: Morgan Kaufmann, 148-156.

Freund, Y. & Schapire, R. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. Computer and System Sciences 55(1), 119-139.

Friedman, J. & Meulman, J. 2002. Clustering objects on subsets of attributes. Department of Statistics, Stanford University, CA, USA. Technical report.

Friedman, J. 1994. Flexible metric nearest neighbour classification. Department of Statistics, Stanford University, CA, USA. Technical report.

Fünkranz, J. 2002. Pairwise classification as an ensemble technique. In T. Elomaa, H. Mannila & H. Toivonen (Eds.) Proc. of Thirteenth European Conference on Machine Learning, LNCS 2430. Berlin Heidelberg: Springer-Verlag, 97-110.

Fuseida, Y. & Satou, K. 1999. Toward a data mining service from large and heterogeneous genome databases in GenomeNet. In K. Asai, S. Miyano & T. Takagi (Eds.) Genome Informatics 10. Tokyo: Universal Academy Press, 304-305.

Fukunaga, K. 1990. Introduction to statistical pattern recognition. San Diego, CA: Morgan Kaufmann.

Gao, K., Khoshgoftaar, T.M. & Napolitano, A. 2011. Impact of data sampling on stability of feature selection for software measurement data. In Proc. of Twenty Third IEEE Int. Conf. on Tools with Artificial Intelligence. Los

Alamitos, CA: IEEE CS Press, 1004-1011.

Garg, A. & Roth, D. 2001. Understanding probabilistic classifiers. In L. De Raedt, & P. Flach (Eds.) Proc. of Twelfth Conference on Machine Learning. LNCS 2167. Berlin Heidelberg: Springer, 179-191.

Gene Expression Mechanism. 2011. Available at [http://www.news-medical.net/health/Gene-Expression-Mechanism.aspx] (refered: December 16, 2011).

Genetics Home reference. 2011. Available at [http://ghr.nlm.nih.gov/handbook/basics/dna] (refered: December 16, 2011).

Ghosh, J. 2002. Multiclassifier systems: Back to the future. In F. Roli & J. Kittler (Eds.) Proceedings of Third Int. Workshop on Multiple Classifier Systems, Vol. 3, LNCS 2364. Berlin Heidelberg: Springer-Verlag, 1-15.

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D. & Lander, E.S. 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286(5439), 531-537.

Guruswami, V. & Sahami, A. 1999. Multiclass learning, boosting, and error-correcting codes. In P. Long & B.D. Shaj (Eds.) Proc. of Twelfth Annual Conference on Computational Learning Theory, New York: ACM Press, 145-155.

Halck, O.M. 2002. Using hard classifiers to estimate conditional class probabilities. In T. Elomaa, H. Mannila & H. Toivonen (Eds.) Proc. of Thirteenth European Conference Machine Learning, Lecture Notes in Artificial Intelligence 2430. Berlin Heidelberg: Springer-Verlag, 123-134.

Hall, M. 1999. Correlation-based feature selection for machine learning. Department of Computer Science, University of Waikato, Hamilton, New Zealand. Ph.D. Thesis.

Hall, M. 2000. Correlation-based feature selection for discrete and numeric class machine learning. In P. Langley (Ed.) Proc. of Seventeenth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 359-366.

Hall, M., Frank, E. Holmes, G., Pfahringer, B., Reutemann, P. & Witten, I.H. 2009. The WEKA data mining software: an update. SIGKDD Explorations 11(1), 10-18.

Halteren, H. 1999. Weighted probability distribution voting, an introduction. In Monachesi, P. (Ed.) Proc. of Tenth Annual Meeting on Computational Linguistics in the Netherlands. Utrecht UILOTS Press, 63-71.

Harries, M. & Horn, K. 1996. Learning stable concepts in domains with hidden changes in context. In M. Kubat & G. Widmer (Eds.) Learning in Context-Sensitive Domains: Workshop Notes, Thirteenth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 106-122.

Hastie, T. & Tibshirni, R. 1996. Discriminant adaptive nearest neighbor classification. IEEE Trans. on Pattern Analysis and Machine Intelligence 18(6), 607-616.

Hastie, T. & Tibshirni, R. 1998. Classification by pairwise coupling. Annals of Statistics 26(2), 451-471.

Ho, T.K. 1998. The Random subspace method for constructing decision forests. IEEE Trans. on Pattern Analysis and Machine Intelligence 20(8). Los Alamitos, CA: IEEE CS Press, 832-844.

Ho, T.K. 2002. A Data complexity analysis of comparative advantages of decision forest constructors. Pattern Analysis and Applications 5, 102-112.

Ho, T.K. 2002. Multiple classifier combination: Lessons and next steps. In A. Kandel & H. Bunke (Eds.) Hybrid Methods in Pattern Recognition, Series in Machine Perception and Artificial Intelligence, Vol. 47. River Edge: World Scientific, 171-198.

Ho, T.K. & Basu, M. 2000. Measuring the complexity of classification problems. In Proc. of Fifteenth Int. Conf. on Pattern Recognition, Vol. 2. Los Alamitos, CA: IEEE CS Press, 43-47.

Ho, T.K. & Basu, M. 2002. Complexity measures of supervised classification problems. IEEE Trans. on Pattern Analysis and Machine Intelligence 24 (3), 289-300.

Ho, T.K., Basu, M. & Law, M.H.C. 2006. Measures of geometrical complexity in classification problems. In M. Basu & T.K. Ho (Eds.) Data Complexity in Pattern Recognition, Advanced Information and Knowledge Processing. London: Springer Verlag, 3-24.

Hong, S.J. 1997. Use of contextual information for feature ranking and discretization. IEEE Trans. on Knowledge and Data Engineering 9(5). Los Alamitos, CA: IEEE CS Press, 718-730.

Hong, S.J., Hosking, J.R.M. & Winograd, W. 1996. Use of randomization to normalize feature merits. In D.L. Dowe, K.B. Korb & J.J. Oliver (Eds.) Proc. of Int. Conf. on Information, Statistics and Induction in Science. Singapore: World Scientific, 10-19.

Hsieh, P.-F. & Landgrebe, D. 1998. Classification of high dimensional data. School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN, USA. Technical report TR-ECE 98-4.

Howe, N. & Cardie, C. 1997. Examining locally varying weights for nearest neighbour algorithms. In D.B. Leake (Ed.) Proc. of Second Int. Conf. on Case-Based Reasoning. New York: Springer-Verlag, 455-466.

Ismail, M.K. & Ciesielski, V. 2003. An empirical investigation of the impact of discretization on common data distributions. In A. Abraham, M. Koppen & K. Franke (Eds.) Proc. of Third Int. Conf. on Hybrid Intelligent Systems: Design and Application of Hybrid Intelligent Systems. Amsterdam: IOS Press, 692-701.

Jimenez, L. & Landgrebe, D. 1999. Supervised classification in high dimensional space: geometrical, statistical, and asymptotical properties of multivariate data. IEEE Trans. on System, Man, and Cybernetics 28(1), 39-54.

John., G., Kohavi, R. & Pfleger, K. 1994. Irrelevant features and subset selection problem. In Cohen, W. & Hirsh, H. (Eds.) Proc. of Eleventh Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 121-129.

Kenney, J.F. & Keeping, E.S. 1962. Linear regression and correlation. In Mathematics of Statistics, 3rd ed. Princeton: Van Nostrand, 252-285.

Kim, W. & Seo, J. 1991. Classifying schematic and data heterogeneity in multidatabase systems. Los Alamitos, CA: IEEE CS Press 24(12), 12-18.

Kira, K. & Rendell, L. 1992a. A practical approach to feature selection. In D. Sleeman & P. Edwards (Eds.) Proc. of Ninth Int. Conf. on Machine Learning. San Mateo, CA: Morgan Kaufmann, 249-256.

Kira, K. & Rendell, L. 1992b. The feature selection problem: Traditional methods and a new algorithm. In Proc. of Tenth National Conference on Artificial Intelligence. Cambridge, MA: MIT Press, 129-134.

Kleinberg, E. 1990. Stochastic discrimination. Annals of Mathematics and Artificial Intelligence 1(1), 207-239.

Kleinberg, E. 2000. On the algorithmic implementation of stochastic discrimination. IEEE Trans. on Pattern Analysis and Machine Intelligence 22 (5), 473-490.

Kohavi, R. 1994. Feature subset selection as search with probabilistic estimates. In R. Greiner (Ed.) AAAI Fall Symposium Series on Relevance. Menlo Park, CA: AAAI Press, 122-126.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In C. Mellish (Ed.) Proc. of Fourteenth Int. Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann, 1137-1145.

Kohavi, R. & John, G. 1998. The wrapper approach. In H. Liu & H. Motoda (Eds) Feature Extraction, Construction and Selection: A Data Mining Perspective. Boston, MA: Kluwer Academic Publishers, 38-50.

Koller, D. & Sahami, M. 1996. Toward optimal feature selection. In L. Saitta (Ed.) Proc. of Thirteenth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 284-292.

Kononenko, I. 1994. Estimating attributes: analysis and extensions of Relief. In F. Bergadano & L. De Raedt (Eds.) Proc. of European Conference on Machine Learning, LNCS 784. Berlin Heidelberg: Springer-Verlag, 171-182.

Kononenko, I. 1995. On biases in estimating multivalued attributes. In Proc. of Fourteenth Int. Joint Conference on Artificial Intelligence. San Mateo, CA: Morgan Kaufmann, 1034-1040.

Kononenko, I. & Hong, S.J. 1997. Attribute selection for modelling. Future Generation Computer Systems 13(2-3), 181-195.

Kononenko, I., Šimec, E. & Robnik-Šikonja, M. 1997. Overcoming the myopia of inductive learning algorithms. Applied Intelligence 7(1), 39-55.

Kowalski, C. J. 1972. A commentary on the use of multivariate statistical methods in anthropometric research. American Journal of Physical Anthropology 36(1), 119-132.

Kriegel, H.-P., Borgwardt, K.M., Kröger, P., Pryakhin, A., Schubert, M. & Zimek, A. 2007. Future trends in data mining. In G. Webb (Ed.) Data Mining and Knowledge Discovery 6(2), Berlin Heidelberg: Springer-Verlag, 87-97.

Kriegel, H.-P., Kröger, P. & Zimek, A. 2009. Clustering high dimensional data: a survey on subspace slustering, pattern-based clustering, and correlation clustering. ACM Trans. on Knowledge Discovery from Data 3(1), 1–58.

Kullback, S. 1968. Information theory and statistics. New York, NY: John Wiley & Sons.

Kuncheva, L. 2004. Combining pattern classifiers: methods and algorithms. New York, NY: John Wiley & Sons.

Kuncheva, L. & Whitaker, C. 2001. Feature subsets for classifier combination: an enumerative experiment. In J. Kittler & F. Roli (Eds.) Proc. of Second Int. Workshop on Multiple Classifier Systems, LNCS 2096. Berlin Heidelberg: Springer-Verlag, 228-237.

Kuncheva, L. & Whitaker, C. 2002. Using diversity with three variants of boosting: aggressive, conservative, and inverse. In J. Kittler & F. Roli (Eds.) Proc. of Third Int. Workshop on Multiple Classifier Systems, LNCS 2364. Berlin Heidelberg: Springer-Verlag, 81-90.

Lang, K.J. & Witbrock, M.J. 1988. Learning to tell two spirals apart. In D. Touretzky, Hinton, G. & Sejnowski, T. (Eds) Proc. of Connectionist Models Summer School. San Mateo, CA: Morgan Kaufmann, 52-59.

Langley, P. 1994. Selection of relevant features in machine learning. In R. Greiner (Ed.) AAAI Fall Symposium Series on Relevance. Menlo Park, CA: AAAI Press, 140-144.

Lazarevič, A. & Obradovič, Z. 2001a. Adaptive boosting techniques in heterogeneous and spatial databases. Intelligent Data Analysis 5(4), 285-308.

Lazarevič, A. & Obradovič, Z. 2001b. Boosting localized classifiers in heterogeneous databases. In Proc. of First SIAM Int. Conf. on Data Mining. Philadelphia, PA: SIAM Press, 540-553.

Lazarevič, A., Fiez, T. & Obradovič, Z. 2000. Adaptive boosting for spatial functions with unstable driving attributes. In Terano, T., Liu, H. & Chen, A.L.P. (Eds.) Proc. of Pacific-Asia Conference on Knowledge Discovery and Data Mining, Lecture Notes in Artificial Intelligence 1805. Berlin: Springer-Verlag, 329-340.

Liang, J., Yang, S. & Winstanley, A. 2008. Invariant optimal feature selection: A distance discriminant and feature ranking based solution. Pattern Recognition 41(5), 1429-1439.

Lindenbaum, M., Markovitch, S. & Rusakov, D. 1999. Selective sampling for nearest neighbor classifiers. In Hendler, J. & Subramanian, D. (Eds.) Proc. of Sixteenth National Conference on Artificial Intelligence and Eleventh Conference on Innovative Applications of Artificial Intelligence. Menlo Park, CA: AAAI/MIT Press, 366-371.

Liu, H. & Motoda, H. 1998. Feature selection for knowledge discovery and data mining. Norwell, MA, USA: Kluwer Academic Publishers.

Lopez de Mántaras, R. 1991. A distance-based attribute selection measure for decision tree induction. Machine Learning 6(1), 81-92.

Maclin, R. & Opitz, D. 1997. An empirical evaluation of bagging and boosting.

In Kuipers, B. & Webber, B.L. (Eds.) Proc. of Fourteenth National Conference on Artificial Intelligence. Menlo Park, CA: AAAI/MIT Press, 546-551.

Madeira, S.C. & Oliveira, A.L. 2004. Biclustering algorithms for biological data analysis: a survey. IEEE/ACM Trans. on Computational Biology and Bioinformatics1(1), 24-45.

Mao, K.Z. & Tang W. 2011. Recursive Mahalanobis separability measure for gene subset selection. IEEE/ACM Trans. on Computational Biology and Bioinformatics 8(1), 266-272.

Masulli, F. & Valentini, G. 2000. Comparing decomposition methods for classification. In Howlett, R.J & Jain L.C. (Eds.) Proc. of Fourth Int. Conf. on Knowledge-Based Intelligent Engineering Systems & Allied Technologies. Piscataway: IEEE CS Press, 788-791.

Mechelen, I., Bock, H.H., De Boeck, P. 2004. Two-mode clustering methods: a structured overview. Statistical Methods in Medical Research 13(5), 363-394.

Michalski, R S., Stepp, R.E. & Diday E. 1981. A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts. In L.N. Kanal & A. Rosenfeld (Eds.) Progress in Pattern Recognition, Vol. 1, New York: North-Holland, 33-56.

Michie, D., Spiegelhalter, D. & Taylor, C. 1994. Machine learning, neural and statistical classification. New York: Ellis Horwood.

Mitchell, T. 1980. The need for biases in learning generalizations. In J. Shavlik & T. Dietterich (Eds.) Readings in Machine Learning. San Mateo, CA: Morgan Kaufmann, 184-191.

Monard, M.C. & Battista, G.E.A.P.A. 2003. Learning with skewed class distributions. Advances in Logic, Artificial Intelligence and Robotics 85(1), 173-180.

Moreira, M. & Mayoraz, E. 1998. Improved pairwise coupling classification with correcting classifiers. In M. Someren & J. Siekmann (Eds.) Proc. of Ninth European Conference on Machine Learning, Lecture Notes in Artificial Intelligence 1398. Berlin Heidelberg: Springer-Verlag, 160-171.

Mukherjee, S., Tamayo, P., Rogers, S., Rifkin, S. Engle, A., Campbell, O., Golub, T.R. & Mesirov J.P. 2003. Estimating dataset size requirements for classifying DNA Microarray data. Journal of Computational Biology 10(2), 119-142.

Nadler, M. & Smith, E.P. 1993. Pattern recognition engineering. New York, NY: John Wiley & Sons, 293-294.

National Institute of General Medical Sciences. 2011. Available at http://publications.nigms.nih.gov/thenewgenetics/chapter1.html#c1] (referred: December 16, 2011).

O'Sullivan, J., Langford, J., Caruana, R. & Blum, A. 2000. FeatureBoost: A meta-learning algorithm that improves model robustness. In P. Langley (Ed.) Proc. of Seventeenth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 703-710.

Opitz, D. & Maclin, R. 1999. Popular ensemble methods: an empirical study. Artificial Intelligent Research 11(1), 169-198.

Orriols-Puig, A., Macià, N. & Ho, T.K. 2010. Documentation for the Data Complexity Library in C++. Ramon Llull University, Barcelona, Spain. Technical report GRSI 2010001.

Opitz, D. 1999. Feature selection for ensembles. In T. Dean & K. McKeown (Eds.) Proc. of Sixteenth National Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press, 379-384.

Oza, N. & Tumer, K. 2001. Input decimation ensembles: Decorrelation through dimensionality reduction. In J. Kittler & F. Roli (Eds.) Proc. of Second Int. Workshop on Multiple Classifier Systems, LNCS 2096, Berlin Heidelberg: Springer-Verlag, 238-247.

Oza, N. & Tumer, K. 1999. Dimensionality reduction through classifier ensembles. Computational Sciences Division, NASA Ames Research Center, Moffett Field, CA, USA. Technical report NASA-ARC-IC-1999-126.

Parmanto, B., Munro, P. & Doyle, H. 1996. Improving committee diagnosis with resampling techniques. In D. Touretzky, M. Mozer & M. Hesselmo (Eds.) Advances in Neural Information Processing Systems, Vol. 8. Cambridge, MA: AAAI/MIT Press, 882-888.

Parsons, L., Haque, E., & Liu, H. 2004. Subspace clustering for high dimensional data: A review. SIGKDD Explorations 6 (1), 90-116.

Pavlidis, P., Weston, J., Cai, J. & Grundy, W.N. 2001. Gene functional classification from heterogeneous data. In T. Gaasterland, P. Karp, K. Karplus, C. Ouzounis, C. Sander & A. Valencia (Eds.) Proc. of Fifth Int. Conf. on Computational Molecular Biology. Menlo Park, CA: AAAI Press, 242-248.

Piatetsky-Shapiro, G. 2007. Data mining and knowledge discovery 1996 to 2005: overcoming the hype and moving from "university" to "business" and "analytics". In G. Webb (Ed.) Data Mining and Knowledge Discovery 6(2), Berlin Heidelberg: Springer-Verlag, 99-105.

Piatetsky-Shapiro, G. & Tamayo, P. 2003. Microarray data mining: facing the challenges. SIGKDD Explorations 5(2), 1-5.

Pierson, W. 1998. Using boundary methods for estimating class separability. Department of electrical Engineering, The Ohio State University. Ph.D. Thesis.

Pinheiro, J.C. & Sun, D.X. 1998. Methods for linking and mining massive heterogeneous databases. In R. Agrawal & P. Stolorz (Eds.) Proc. of Fourth Int. Conf. in Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI Press, 309-313.

Pranckeviciene, E., Ho, T.K., & Somorjai, R. 2006. Class separability in subspaces reduced by feature selection. In: Proc. of Eighteenth Int. Conf. on Pattern Recognition, Vol. 3. Washington, DC: IEEE CS Press, 254-257.

Prodromidis, A. & Stolfo, S. 1998. Pruning classifiers in a distributed meta-learning system. Department of Computer Science, Columbia University, New York, NY. Technical Report CUCS-011-98.

Puuronen S., Skrypnyk I. & Tsymbal A. 2001. Ensemble feature selection based on contextual merit and correlation heuristics. In A. Caplinskas & J. Eder (Eds.) Proc. of Fifth East-European Conference on Advances in Databases and Information Systems, LNCS, Vol. 2151. Berlin Heidelberg: Springer-Verlag, 155-168.

Quinlan. J. 1986. Induction of decision trees. Machine Learning 1(1), 81-106.

Quinlan. J. 1993. C4.5 programs for machine learning. San Mateo, CA: Morgan Kaufmann.

Ramaswamy, S., Tamayo, P., Rifkin, R., Mukherjee, S., Yeang, C.-H., Angelo, M., Ladd., C., Reich, M., Latulippe, E., Mesirov, J.P., Poggio, T., Gerald, W., Loda, M., Lander, E.S., & T.R. Golub. 2001. Multiclass cancer diagnostics using tumor gene expression signatures. In Proc. of National Academy of Sciences of the United States of America, Medical Sciences, 98(26). PNAS Press, 15149–15154.

Reich, M., Ohm, K., Tamayo, P., Angelo, M., & Mesirov, J.P. 2004. GeneCluster 2.0: An advanced toolset for bioarray analysis. Bioinformatics 20(11), 1797-1798.

Ricci, F. & Aha, D. 1997a. Extending local learners with error-correcting output codes. Naval Research Laboratory, Navy Center of Applied Research in Artificial Intelligence, Washington DC. Technical report AIC-97-001.

Rissanen, J. 1978. Modeling by shortest data description. Automatica 14(1), 465-471.

Robnik-Šikonja, M. & Kononenko, I. 1996. Context-sensitive attribute estimation in regression. In M. Kubat & G Widmer (Eds.) Learning in Context-Sensitive Domains: Workshop Notes, Thirteenth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 43-52.

Robnik-Šikonja, M. & Kononenko, I. 1999. Attribute dependencies, understandability and split selection in tree based models. In I. Bratko & S. Dzeroski (Eds.) Proc. of Sixteenth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 344-353.

Robnik-Šikonja, M. & Kononenko, I. 2003. Theoretical and empirical analysis of ReliefF and RReliefF. Machine Learning 53(1-2), 23-69.

Rosmalen, J., Groenen, P.J.F. & Trejos, J. 2009. Optimization Sstrategies for two-mode partitioning. Journal of Classification 26(2), 155-181.

Sahami, M. 1996. Learning limited dependence Bayesian classifiers. In E. Simoudis, J. Han & U.M. Fayyad (Eds.) Proc. of Second Int. Conf. on Knowledge Discovery and Data Mining. Menlo Park, CA: AAAI Press, 335-338.

Salzberg, S. 1991. A nearest hyperrectangle learning method. Machine Learning 6(3), 277-309.

Sarvestani, A.S., Safavi, A.A., Parandeh, N.M. & Salehi, M. 2010. Predicting breast cancer survivability using data mining techniques. In Proc. of Second Int. Conf. on Software Technology and Engineering, Los Alamitos, CA: IEEE CS Press, 227- 231.

Schaffer, C. 1993. Selecting a classification method by cross-validation. Machine

Learning 13(1), 135-143.
Schaffer, C. 1994. A conservation law for generalization performance. In W. Cohen & H. Hirsh (Eds.) Proc. of Eleventh Int. Conf. on Machine Learning. New Brunswick, NJ: Morgan Kaufmann.
Schapire, R., Freund, Y., Bartlett, P. & Lee, W. S. 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. The Annals of Statistics 26(5), 1651-1686.
SEER. 2008. Surveillance, Epidemiology, and End Results (SEER) Program (www.seer.cancer.gov) Limited-Use Data (1973-2006), National Cancer Institute, DCCPS, Surveillance Research Program, Cancer Statistics Branch, released April 2009, based on the November 2008 submission.
SEER: Measures of Cancer Survival. 2011. Surveillance research guidelines from National Cancer Institute. Available at [http://surveillance.cancer.gov/survival/measures.htm] (refereed: December 10, 2011).
Singh, S, Singh, M. & Markou, M. 2002. Feature selection for face recognition based on data partitioning. In Proc. of Sixteenth Int. Conf. on Pattern Recognition, Vol. 1, Los Alamitos, CA: IEEE CS Press, 680-683.
Skrypnyk I. 2002a. Comparison of feature selection strategies for hearing impairments diagnostics. In Proc. of Fifteenth IEEE Symposium on Computer-Based Medical Systems. Los Alamitos, CA: IEEE CS Press, 25-30.
Skrypnyk I. 2002b. Neural networks for analyzing rehabilitation of hearing impairments in children via voice descriptors. In Wani, M.A., Arabnia, H.A., Cios, K.J., Hafeez, K. & Kendall, G. (Eds.) Proc. of First Int. Conf. on Machine Learning and Applications. Los Alamitos, CA: IEEE CS Press, 20-26.
Skrypnyk I. 2004. Exploring classification heterogeneity with IPA. In S. Draghici, T.M. Khoshgoftaar, V. Palade, W. Pedrycz, M.A. Wani & X. Zhu (Eds.) Proc. of Ninth IEEE Int. Conf. on Machine Learning and Applications. Los Alamitos, CA: IEEE CS Press, 272-279.
Skrypnyk, I. 2005. Combining class encoding and local feature selection for class heterogeneity decomposition. Department of Computer Science and Information Systems, University of Jyväskylä. Ph.Lic. thesis.
Skrypnyk I. 2007. Local selective partitioning for heterogeneous classification problems. In R. Stahlbock, S.F. Crone & S. Lessmann (Eds.) Proc. of Third Int. Conf. on Data Mining. Las Vegas, NV: CSREA Press, 84-90.
Skrypnyk I. 2008. Feature weighting to improve class discrimination in subspaces. In R. Stahlbock, S.F. Crone & S. Lessmann (Eds.) Proc. of Fourth Int. Conf. on Data Mining. Las Vegas, NV: CSREA Press, 58-64.
Skrypnyk I. 2009. Generation of ”weak” models in stochastic discrimination, In R. Stahlbock, S.F. Crone & S. Lessmann (Eds.) Proc. of Fifth Int. Conf. on Data Mining, Las Vegas, NV: CSREA Press, 368-374.
Skrypnyk, I. 2010. Bidirectional subspace decomposition in classification. In R. Stahlbock, S.F. Crone, M. Abou-Nasr, H.R. Arabnia, N. Kourentzes, P.

Lenca, W.-M. Lippe & G.M. Weiss (Eds.) Proc. of Sixth Int. Conf. on Data Mining. Las Vegas, NV: CSREA Press, 256-266.

Skrypnyk, I. 2011. Irrelevant features, class separability, and complexity of classification problems. In Proc. of Twenty Third IEEE Int. Conf. on Tools with Artificial Intelligence. Los Alamitos, CA: IEEE CS Press, 998-1003.

Skrypnyk, I. & Ho, T.K. 2003. Feature selection and training set sampling for ensemble learning on heterogeneous data. Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University, NJ. Technical report 2003-23.

Skrypnyk I. & Ho T.K. 2006. Hyper-rectangular and *k*-Nearest-Neighbour models in stochastic discrimination. S.F. Crone, S. Lessmann & R. Stahlbock (Eds.) Proc. of Second Int. Conf. on Data Mining. Las Vegas, NV: CSREA Press, 57-63.

Skrypnyk, I., Puuronen, S. & Tsymbal, A. 2003. Combination of feature selection -based spacing and boosting for more accurate ensemble learning. Department of Computer Science and Information systems, University of Jyväskylä, Finland. Unpublished manuscript.

Skurichina, M. 2001. Stabilizing weak classifiers. Department of Imaging Science and Technology, Delft University of Technology. Ph.D. thesis.

Skurichina, M. & Duin, R. 2001. Bagging and random subspace method for redundant feature spaces. In J. Kittler & F. Roli (Eds.) Proc. of Second Int. Workshop on Multiple Classifier Systems. LNCS 2096. Berlin Heidelberg: Springer-Verlag, 1-10.

Soofi, E.S. 2000. Principal information: theoretic approaches. Journal of the American Statistical Association 95(1), 1349-1353.

Stanfill, C. & Waltz, D. 1986. Toward memory-based reasoning. Communications of the ACM 29(12), 1213-1228.

Starck, J.-L., Murtagh, F. & Bijaoui, A. 1998. Image processing and data analysis: the multiscale approach. London: Cambridge University Press.

Su, Y., Murali, T.M., Pavlovic, V., Schaffer, M. & Kasif, S. 2003. Rankgene: a program to rank genes from expression data. Bioinformatics 19(12), 1578-1579.

Tamayo, P., Reich, M., Ohm, K. Subramanian, A., Ross, K., Ramaswamy, S., Jill Mesirov, J., Golub, T.& Angelo M. 2002. Molecular pattern recognition with GeneCluster 2: tutorial and examples. Available at http://www.broadinstitute.org/cancer/software/genecluster2/gc2.html (refereed: December 16, 2011).

Thierry-Mieg, N. 2000. Protein-protein interaction prediction for C. elegans. In Proc. of Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases, Workshop on Knowledge Discovery in Biology. Berlin Heidelberg: Springer-Verlag.

Toussaint, G. 2002. Proximity graphs for nearest neighbor decision rules: Recent progress. In Wegman, E.J. & Braverman, A. Proc. of Thirty Fourth INTERFACE Symposium, Computing Science and Statistics, Vol. 34.

Tsymbal, A. 2002. Dynamic integration of data mining methods in knowledge

discovery systems. Department of Computer Science and Information Systems, University of Jyv□skyl□. Ph.D. thesis.

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

Tumer, K. & Ghosh, J. 1996a. Classifier combining: analytical results and implications. In Proc. of AAAI'96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms. Cambridge, MA: AAAI/MIT Press, 126-132.

Tumer, K. & Ghosh, J. 1996b. Error correlation and error reduction in ensemble classifiers. Connection Science 8(3-4), Special Issue on Combining Artificial Neural Networks: Ensemble Approaches, 385-404.

Turney, P. 1993. Robust classification with context-sensitive features. In P. Chung, G.L. Lovegrove & M. Ali (Eds.) Proc. of Sixth Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems. London: Gordon and Breach Publishing, 268-276.

Turney, P. 1996. The identification of context-sensitive features: A formal definition of context for concept learning. In M. Kubat & G Widmer (Eds.) Learning in Context-Sensitive Domains: Workshop Notes, Thirteenth Int. Conf. on Machine Learning. San Francisco, CA: Morgan Kaufmann, 53-59.

Tversky, A. 1977. Features of similarity. Psychological Review 84(4), 327-352.

Valentini, G. & Masulli, F. 2002. Ensembles of learning machines. In M. Marinaro & R. Tagliaferri (Eds.) Neural Nets, Proc. of Thirteenth Italian Workshop on Neural Nets, LNCS 2486. Berlin Heidelberg: Springer-Verlag, 3-22.

Ventura, D. & Martinez, T. 1995. An empirical comparison of discretization methods. In Proc. of Tenth Int. Symposium on Computer and Information Sciences, 443-450.

Wang, H. & Khoshgoftaar, T.M. 2011. Measuring stability of threshold-based feature selection techniques. In Proc. of Twenty Third Int. Conf. on Tools with Artificial Intelligence. Los Alamitos, CA: IEEE CS Press, 986-993.

Wang, H.-Q., Wong, H.-S., Huang, D.S. & Shu, J. 2007. Extracting gene regulation information for cancer classification. Pattern Recognition 40(1), 3379-3392.

Webb, G. 2000. MultiBoosting: A technique for combining boosting and wagging. Machine Learning 40(2), 159-196.

Weiss, G. & Provost, F. 2003. Learning when training data are costly: the effect of class distribution on tree induction. Journal of Artificial Intelligence Research 19(1), 315-354.

Weiss, S. & Indurkhya, N. 1998. Predictive data mining: A practical guide. San Francisco, CA: Morgan Kaufmann.

Wettscherech, D. 1994. A study of distance-based machine learning algorithms. Department of Computer Science, Oregon State University, OR, USA. Ph.D. Thesis.

White, A.P. & Liu, W.Z. 1994. Bias in information-based measures in decision

tree induction. Machine Learning 15(3), 321-329.

Witten, I.H. & Frank, E. 2005. Data mining: practical machine learning tools and techniques, 2nd ed. San Francisco, CA: Morgan Kaufmann.

Widdows, D. 2003. Geometry and meaning. Chicago, IL: University of Chicago Press.

Wilson, R.D. & Martinez, T.R. 1997. Improved heterogeneous distance functions. Journal of Artificial Intelligent Research 6(1), 1-34.

Windeatt, T. & Ardeshir, G. 2002. Boosted tree ensembles for solving multiclass probems. In J. Kittler & F. Roli (Eds.) Proc. of Third Int. Workshop on Multiple Classifier Systems, LNCS 2364. Berlin Heidelberg: Springer-Verlag, 42-51.

Wolpert, D. 1992. Stacked generalization. Neural Networks 5(2), 241-259.

Wu, X., Kumar, V., Quinlan, J.R., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G.J., Ng, A., Liu, B., Yu, P.S., Zhou, Z.-H., Steinbach, N., Hand, D.J. & Steinberg, D. 2008. Top 10 algorithms in data mining. Knowledge Information Systems 14(1), 1-37.

Xiang, S., Nie, F. & Zhang, C. 2008. Learning a Mahalanobis distance metric for data clustering and classification. Pattern Recognition 41(12), 3600-3612.

Ye, J., Zhao, Z. & Liu, H. 2007. Adaptive distance metric learning for clustering. In Proc. of Twentieth Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE CS Press, 1-7.

You, D. & Martinez, A.M. 2010. Bayes optimal kernel discriminant analysis. In Proc. of Twenty Third Conference on Computer Vision and Pattern Recognition. Los Alamitos, CA: IEEE CS Press, 3533–3538.

Yu, L. & Liu, H. 2003. Feature selection for high-dimensional data: A fast correlation-based filter solution. In T. Fawcett & N. Mishra (Eds.) Proc. of Twentieth Int. Conf. on Machine Learning. Menlo Park, CA: AAAI Press, 856-863.

Zaffalon, M. & Hutter, M. 2002. Robust feature selection by mutual information distributions. In A. Darwiche & N. Friedman (Eds.) Proc. of Eighteenth Int. Conf. on Uncertainty in Artificial Intelligence. San Francisco, CA: Morgan Kaufmann, 577-584.

Zheng, Z. & Webb, G. 1998. Multiple boosting: A combination of boosting and bagging. In Proc. of Fourth Int. Conf. on Parallel and Distributed Processing Techniques and Applications. Las Vegas, NV: CSREA Press, 1133-1140.

Zheng, Z., Webb, G. & Ting, K.M. 1998. Integrating boosting and stochastic attribute selection committees for further improving the performance of decision tree learning. In Proc. of Tenth IEEE Int. Conf. on Tools with Artificial Intelligence. Los Alamitos, CA: IEEE CS Press, 216-223.

Appendix 1

The basic learning algorithms for classification

The state-of-the-art learning algorithms described in this section represent different approaches to learning. These algorithms are well known in data mining community and have proven effective in practice. J48 (C4.5) decision tree represents an information-theoretic approach, *k*-Nearest Neighbor represents a distance-based approach, and Naïve Bayes exemplifies a probabilistic approach to learning. There are other learning algorithms originated from pattern recognition, statistics and artificial intelligence, including Fisher's linear/quadratic/logistic discriminant, association rules, rough sets, neural networks, support vector machines/kernel methods, but their performance is out of scope of this thesis.

Of particular interest here is performance of J48 (C4.5) decision tree, *k*-Nearest Neighbor, and Naïve Bayes on heterogeneous classification problems with locally relevant and interacting features, and as the component classifiers of ensemble generated on locally relevant features and subsets of instances representing homogeneous regions.

1.1 J48 (C4.5) decision tree

Decision tree is a widely used learning algorithm that has proved to be effective in many practical tasks (Michie *et al*., 1994). A decision tree is represented as a set of *nodes* with incoming and outgoing *branches*, where nodes correspond to features, and branches correspond to their associated values. A

node from which branches merely outgo is called a *root*, a node to with branches merely income is called a *leaf*. Leaves in a decision tree correspond to class values. Together a node and the outgoing branches represent a decision about the path an instance follows when being classified by the tree.

A set of instances (usually a subset of the original instance set in the tree generation process) generally contains instances from different classes. If a subset contains instances of one class only, no more tests need to be applied to the decision path that led to separation of the subset from its superset. The degree of uncertainty about the classes of instances in a set is often called its class *impurity*.

Given a set of training instances, a decision tree is usually induced by repeatedly dividing instances according to their values for a particular feature. This is known as a "divide and conquer" or "recursive partitioning" approach to learning (Breiman *et al*., 1984). All instances belong to one part in the first 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 leaf nodes belong to the same class.

Different decision tree learning algorithms use different ways to make splits and grow the tree. The simplest of the growing schemes used in the ID3 and C4.5 algorithms (Quinlan, 1986; Quinlan, 1993) proceeds from the tree growing in a top-down fashion. In order to classify an instance different values of a root feature are tested for a split. Various splitting functions have been investigated, for example, in Ho (1998).

The impurity measure originally used in ID3 (Quinlan, 1986) to evaluate a partition induced by current split is Information gain that is based on entropy calculation (Formula 23, Subsection 4.2.1). However, it has one significant drawback: it does not take into account the number of feature values. As a result, this leads to a bias towards features taking many values overestimating them.

The C4.5 decision tree learning algorithm (Quinlan, 1993) has a number of improvements over ID3. In C4.5 Gain ratio (Formula 24, Subsection 4.2.1) is used instead of Information gain. The Gain ratio measure compensates for the number of features normalizing by the information encoded in the split itself. The other extensions of C4.5 include, for example, ability to deal with the missing feature values estimating probabilities of various possible results of the split.

Both ID3 and C4.5 handle continuous features in the following way. In the case of a continuous feature f_j all T values it takes on the training set TR are considered in an increasing order, $v_{1,j}, \ldots, v_{t,j}, \ldots, v_{T,j}$.Then, for a value $v_{t,j}$, $t = 1 \ldots T$, instances are partitioned into subsets where f_j takes values up to and including $v_{t,j}$. For each of these splits Information gain or Gain ratio is computed, and the split that maximizes the gain is chosen.

The decision tree built using the training set, because of the way it was built, deals correctly with most of the instances in the training set. In fact, in

order to do so, it may become quite complex, with long and very uneven paths. Often this leads to overfitting.

Pruning is a common strategy to avoid overfitting for decision tree learning. It is performed by replacing a whole sub-tree by a leaf node. Decision trees may be easily interpreted in the form of logical decision rules. Then, the replacement takes place if a decision rule establishes that the expected error rate in the sub-tree is greater than in a single leaf. (Quinlan, 1993)

Pruning improves generalization performance on a relatively small set of pruning validation instances. In C4.5 pruning is performed using the upper bound of a confidence interval on the resubstitution error as the error estimate. Nodes with fewer instances having a wider confidence interval are removed if the difference in error between them and their parent nodes is not significant. (Quinlan, 1993)

Determining the relative importance of features is basic to all decision tree algorithms. While constructing a tree at each interior node a feature to make a split on is selected and the instance set is divided into subsets. However, each time the best feature is selected for splitting the test is done relying on this individual feature's effect on class discrimination. This is problematic especially at the nodes near the root, because the context of other features is ignored (Hong, 1997).

In the decision tree algorithms based on impurity measures instances with different class values are separated into different sub-trees. In Robnik-Šikonja and Kononenko (1999) it is stated that strong class-conditional dependencies (interactions) between features are not properly detected by these measures, therefore dependent features are not selected as splits near the root of the tree, which would guarantee compact representation of dependencies. Instead interacting features are selected in subsequent levels of the tree, mostly near the fringe, which causes node replication, and might decrease accuracy.

Langley and Sage (1997) have shown that the effect of irrelevant features on C4.5 depends on the classification problem. As long as relevant features discriminate between classes being independent on other features, accuracy of C4.5 is unaffected by introduction of irrelevant features. However, when features interact (as in the parity-like problems), that is none of the relevant features in isolation is able to discriminate classes, accuracy of C4.5 severely degrades after addition of irrelevant features.

How deeply should a decision tree be developed? Too shallow tree growth will smooth out the boundaries between classes, thus generating a stable but rigid and therefore, inaccurate model. On the other hand, an overgrown tree will build overdetailed boundaries that will be mostly determined by the random nature of the data. Consequently, class boundaries will then heavily depend on the particular sample at hand, and the predictions of the model will be unreliable.

1.2 Naïve Bayes

The Bayesian learning algorithm, often called Naïve Bayes, represents an

approach to construction of a predictive model based on the Bayes theorem. This learning algorithm is originated from pattern recognition (Duda & Hart, 1973). Contrary to the C4.5 and *k*-Nearest Neighbor learning algorithms (Subsection 2.3.3), Naïve Bayes does not assume a deterministic relationship between each instance and its class. In many real-world classification problems there is no deterministic relationship, because all relevant information is not encoded in the input representation, or it may be due to real randomness of the problem domain (Halck, 2002). Instead, a probability of class membership called conditional class probability is estimated.

Since the underlying probabilities of class membership are always unknown, the probabilities are estimated from the training set. The posterior probability of each class is calculated, given the feature values present in the instance, and then the instance is assigned to the class with the highest probability. This learning algorithm produces a linear decision boundary through the instance space. Each time the algorithm encounters a new instance from the training set, the probabilities stored with the specified class are updated. Upon being given an unclassified instance from the test set, the classifier created uses an evaluation function to rank the alternative classes based on their probabilistic summaries, and assigns the instance to the class with the highest score (Langley *et al.*, 1992).

Formula 67 shows a simplified Bayes formula used in the Naïve Bayes learning algorithm, which makes an assumption that feature values are statistically independent within each class (Duda *et al.*, 2001; Hall, 1999).

$$P\left(y_i = c_d \middle| x_{i,1},\, x_{i,2},\, \ldots,\, x_{i,N}\right) = \frac{P\left(y_i = c_d\right) \prod_{j=1}^{N} P\left(x_{i,j} \middle| y_i = c_d\right)}{P\left(x_{i,1},\, x_{i,2},\, \ldots,\, x_{i,N}\right)} \tag{67}$$

The left side of Formula 1 is the posterior probability of class c_d, $d = 1 \ldots D$, given the feature values, $x_{i,1}, x_{i,2}, \ldots, x_{i,N}$, observed in the instance to be classified. The denominator on the right side of Formula 1 is constant and can be omitted. The remaining probabilities can be calculated from the training set. The posterior probability of each feature value $x_{i,j}$ given the class value c_d can be defined as shown in Formula 68.

$$P\left(x_{i,j} | y_i = c_d\right) = \frac{P\left(x_{i,j}\right) P\left(y_i = c_d | x_{i,j}\right)}{P\left(y_i = c_d\right)} \tag{68}$$

Then Formula 67 can be rewritten as follows.

$$P\left(y_i = c_d | x_{i,1}, x_{i,2}, \ldots, x_{i,N}\right) \propto P\left(y_i = c_d\right) \prod_{j=1}^{N} \frac{P\left(y_i = c_d | x_{i,j}\right)}{P\left(y_i = c_d\right)} \tag{69}$$

This interpretation of the Bayes formula considered in Hong *et al.* (2002) presents a contribution of individual features in calculation of the class probability: the class probability given the values of a set of features is proportional to the prior probability of the class adjusted multiplicatively by factors, each reflecting the influence of the particular feature.

Recent extensions of the Bayesian learning algorithm are complex and not so easily amenable to analysis. But despite its simplicity, the Naïve Bayes learning algorithm performs well on most classification tasks, and is often more accurate than more sophisticated methods (Langley *et al*., 1992). Although the probability estimates that it produces can be inaccurate, it often assigns maximum probability to the correct class (Eibe *et al*., 2000).

An assumption of class-conditional independence between features that Naïve Bayes relies on is rarely valid in practical learning problems: features used for deriving a prediction are not independent of each other, given the predicted class value. However, it has been shown that Naïve Bayes is surprisingly robust to obvious violation of this independence assumption, yielding accurate classification results even when there are clear conditional dependencies. (Hong *et al*., 2002)

Many studies of the Bayesian learning algorithm's performance have been undertaken. For example, in Domingos and Pazzani (1997) theoretical conditions of Naïve Bayes optimality are explored considering the situation when the independence assumption may not hold. In Garg and Roth (2001) the dependence between the number of all joint distributions and the product distribution of Naïve Bayes is established explaining the power of Naïve Bayes beyond the independence assumption. A comparison of the simple Bayesian learning algorithm with state-of-the-art learning algorithms on standard benchmark datasets has been performed in Domingos and Pazzani (1997).

1.3 *k*-Nearest Neighbor

The *k*-Nearest Neighbor learning algorithm is one among the most successful methods for many classification problems. It is the earliest nonparametric method proposed for classification and has been extensively studied in pattern recognition and applied statistics (Duda & Hart, 1973; Dasarathy, 1991). This learning algorithm is based on a similarity concept.

The *k*-Nearest Neighbor learning algorithm, or its variant, an instance-based learning algorithm (Aha *et al*., 1991), classifies an unknown instance from the test set to the plurality class of its *k* nearest neighbors from the training set using some distance metric defined in the feature space, most commonly, Euclidian metrics. The metric chosen to define the distance can strongly affect performance. The optimal choice depends on the classification problem specifics characterized by the respective class distributions in the feature space and within a given problem, on the location of the unknown instance in this feature space (Friedman, 1994).

The decision boundary of the nearest neighbor rule consists of the boundaries of the Voronoi regions that separate the regions of different classes (Toussaint, 2002).

Prediction for a new unclassified instance is based on the closest instance, or several instances (Aha *et al*., 1991; Wettscherech, 1994). To classify a new instance, its distance to all the training instances is calculated and the class label corresponding to the closest training instance is assigned to this instance. A

more sophisticated version of the Nearest Neighbor learning algorithm returns the most frequent class among the *k* closest training instances (denoted *k*-NN) (Aha *et al.*, 1991).

Selection of a distance metric for *k*-Nearest Neighbor is crucial. In Stanfill and Waltz (1986) a value difference metric that can be used for categorical features is described, for symbolic features a Hamming distance may be applied (Dasarathy, 1991; Bay, 1998).

In Boolean domains a natural measure for the *k*-Nearest Neighbor learning algorithm is the number of feature values that differ between the test instance and the stored instance, called *city-block* metric (Langley & Sage, 1997).

k-Nearest Neighbor is known to be very sensitive to irrelevant features (Duda & Hart, 1973, Dasarathy, 1991).

(Blayo *et al.*, 1995) provide more information on Nearest Neighbor performance and error estimates.

1.4 Classifier performance evaluation

Whether various assumptions about the data hold in practice is a nontrivial question. Therefore, applicability of a certain technique is often verified by the classification performance, which is usually evaluated by classification accuracy. Classification performance mainly depends on intrinsic problem complexity, training set size, dimensionality, and type of discriminating function used in a classifier. Intrinsic problem complexity and unambiguity assessment is formalized as a Bayes minimum error, which is estimated using class separability measures.

The results for a classifier are usually summarized in a confusion matrix shown in Table 25, where a, b, c and d represent the number of instances falling into each possible outcome. Therefore, (c+d) is a number of instances in class denoted as positive; (a+b) is a number of instances in class denoted as negative. In multiclass case a negative class includes all classes but positive.

TABLE 25 Confusion matrix for classifier's performance evaluation.

True Class Label	**Predicted class label**	
	Negative Class	Positive Class
Negative Class	a (true negative)	b (false positive)
Positive Class	c (false negative)	d (true positive)

Commonly, the results are evaluated according to the following performance measures easily extended for multiclass problems. True positives rate (TPR) is a ratio of correctly predicted positives to the number of all correctly predicted instances, TPR = d/(a+d). For example, it's a ratio of sick patients correctly diagnosed as sick to all patients with correct diagnoses, both sick and healthy. False positives rate (FPR) is a ratio of negatives predicted as positives to all

incorrectly predicted instances, FPR = c/(c+b). For example, it's a ratio of healthy patients incorrectly identified as sick to all incorrectly diagnosed patients.

The recall (REC) measure (also called sensitivity) is a ratio of true positives per instances of positive class d/(c+d). *Sensitivity* measures the proportion of actual positives which are correctly identified as such (for example, the percentage of sick patients who are correctly identified as having the condition). *Specificity* measures the proportion of negatives which are correctly identified, a/(a+b), for example, the percentage of healthy people who are correctly identified as not having the condition. These two measures are closely related to the concepts of type I and type II errors in statistical tests. A type I error, also known as a false positive, occurs when a statistical test rejects a true null hypothesis. A type II error, a false negative, occurs when the test fails to reject a false null hypothesis.

The precision measure (PRE) is a ratio of true positives per all instances classified as positives d/(b+d). For example, it's a ratio of sick patients correctly diagnosed as sick to all patients diagnosed as sick, including healthy patients. This measure characterizes proportion of actual positives in the population being recognized as such rather than being a characteristic of a classifier.

When dealing with highly imbalanced classes, precision against recall (PR) curve is more informative with respect to algorithm's performance then commonly used Receiver Operator Characteristic (ROC) curve that is true positive rate against false positive rate (Davis & Goadrich, 2006).

F-Measure (F) with respect to a particular class is a harmonic mean of precision and recall, where they are equally weighted, F=(2*REC*PRE)/(REC+PRE).

The balanced error rate (BER) is the average of the errors on each class: BER = 0.5*(b/(a+b) + c/(c+d)).

The area under curve (AUC) is defined as the area under the ROC curve. This area is equivalent to the area under the curve obtained by plotting a/(a+b) against d/(c+d) for each confidence value, starting at (0,1) and ending at (1,0). The area under this curve is calculated using the trapezoid method. In the case when no confidence values are supplied for the classification the curve is given by {(0,1),(d/(c+d),a/(a+b)),(1,0)} and AUC = 1 - BER.

When comparing accuracy of two classifiers statistical significance testing is used. McNemar test (Dietterich, 1998) is used for tests whether combinations of values between two dichotomous variables are equally likely. The output includes a *cross-tabulation* table for each pair and a *test statistics* table for all pairs, showing the number of valid cases, chi-square, and probability for each pair.

1.5 Biases of learning algorithms

Every learning, feature selection, transformation or discretization method, same as other techniques used in the knowledge discovery process, has own bias. A bias is "a rule or method that causes an algorithm to choose one generalized

output over another" (Mitchell, 1980). The bias is influenced by different factors, for example, by the choice of distance function, evaluation function, the basic assumption abound which the method is designed. A learning algorithm must have a bias in order to generalize and it has been shown that no learning algorithm can generalize more accurately than any other when summed over all possible domains (Schaffer, 1994), unless some domain knowledge about the problem is available. It follows then that no feature selection method can be strictly better considering all possible problems with equal probability.

Known biases of learning and feature selection techniques considered in this thesis are mentioned in each particular technique's description. The results obtained in the experimental sections are interpreted taking into account data characteristics and geometric complexity as well as biases of the applied techniques.

Appendix 2

Clustering algorithms

Below a brief introduction to the clustering techniques used in Bidirectional Data Partitioning (BDP) is provided. Both, *k*-Means and DBSCAN are well known algorithms extensively studied in the literature. Therefore, this overview mainly emphasizes their advantages and disadvantages crucial to understand performance of BDP.

2.1 *k*-Means and DBSCAN

k-Means is one of the most commonly used clustering algorithm, but it does not perform well on data with outliers or with clusters of different sizes or non-globular shapes. The single link agglomerative clustering method is the most suitable for capturing clusters with non-globular shapes, but this approach is very sensitive to noise and cannot handle clusters of varying density. Other agglomerative clustering algorithms, e.g., complete link and group average, are not as affected by noise, but have a bias towards finding globular clusters. More recently, clustering algorithms have been developed to overcome some of these limitations. In particular, for low dimensional data, DBSCAN have shown good performance. In DBSCAN, the density associated with a point is obtained by counting the number of points in a region of specified radius, ε, around the point. Points with a density above a specified threshold, MinPoints, are classified as core points, while noise points are defined as non-core points that don't have a core point within the specified

radius. Noise points are discarded, while clusters are formed around the core points. If two core points are within a radius of ε of each other, then their clusters are joined. Border points, which are non-noise and non-core points, are assigned to the clusters associated with any core point within their radius. Thus, core points form the skeleton of the clusters, while border points flesh out this skeleton. While DBSCAN can find clusters of arbitrary shapes, it cannot handle data containing clusters of differing densities, since its density-based definition of core points cannot identify the core points of varying density clusters. (Ertöz *et al*., 2003)

Clustering methods have no access to class label information. Therefore, a good distance metric is crucial for clustering high-dimensional data. A distance metric can be adaptively learned by a clustering algorithm. Most distance-based clustering algorithms appeal to projecting observed data onto a low-dimensional manifold, where geometric relationships such as local or global pairwise distances are preserved. (Ye *et al*., 2007; Chen *et al*., 2007)

2.2 Measuring similarity and distance

Data analysis for supervised and unsupervised learning involves various approaches, such as probability-based and statistical approaches, information-theoretic approach, neural networks, evolutionary algorithms and other. One of the most popular approaches is based on a similarity concept, which requires measuring *similarity* between two instances for learning and prediction.

Measuring similarity involves investigation of a relationship between two instances and specification of a *distance* - a number that is assigned to a pair of instances (points in the feature space), which indicates how far those points are from one another. A distance function is called a *metric* when it is always positive (except when measuring the distance from any points to itself, which must be zero), if it is always symmetric, and if it permits no "short-cuts" or "wormholes". A similarity measure is the converse of a distance function. Similarity functions take a pair of points and return a large similarity measure for the nearby points, a small similarity value for distant points. (Widdows, 2003)

Many techniques used in data mining, including classification, clustering, feature selection/extraction, and constituent measures are very sensitive to the choice of an appropriate distance metric. Considering classification and clustering tasks in a single framework, data projections and metric help to identify data structure. Class labels introduce additional information on data structure, and often given a top priority. Then similarity definition is acquired from class labels by means of metric learning. A distance function used contributes to a success of learning. In clustering, similarity is solely defined by a distance function. If clustering is performed in the presence of some background knowledge or supervisory information (semi-supervised clustering), this information is often expressed as pairwise similarity or

dissimilarity constraints and a chosen metric is adjusted accordingly. Several distance functions popular in data mining are considered below.

Distance functions depend on the working space, so one may choose different ways to determine distance that are appropriate for different applications. In machine learning this choice depends on the problem domain, and hence, on the type of features and their measurement scale. The related data preprocessing issues are considered in Section 7.1.

Majority of metrics are designed to handle continuous features well, but they do not handle discrete and nominal features appropriately. There are also distance functions designed to find reasonable distance values between nominal (symbolic) features, such as the Value Difference Metric (VDM) (Stanfill & Waltz, 1986). However, they largely ignore continuous features requiring discretization to map continuous values into discrete values, which sometimes can degrade generalization accuracy (Ventura & Martinez, 1995).

Heterogeneous Value Difference Metric (HVDM) (Wilson & Martinez, 1997) is a combination of normalized Euclidean distance (divided by the range of feature's values) for continuous features and VDM for nominal features.

The choice of a distance function depends on the type of a problem. In general, some distance functions can be more preferable in the particular domains that the others leading to the prediction accuracy increase. Categorization of the problems where a particular distance function is better than another is yet to be done by the machine learning and pattern recognition research communities.

A variety of distance functions have been developed, including the Minkowsky (Batchelor, 1978), Mahalanobis (Nadler & Smith, 1993), Canberra, Chebychev, Quadratic, Correlation, and Chi-square distance metrics (Michalski *et al*., 1981; Diday, 1974); the Context-Similarity measure (Biberman, 1994); the Contrast Model (Tversky, 1977); hyperrectangle distance functions (Salzberg, 1991; Domingos, 1995) and others. Many of them were developed by the statistical community serving particular goals, like outliers detection. Some measures used in data mining are briefly described below.

Minkowsky distance is the generalized metric distance. When $\lambda = 1$ it becomes Manhattan distance and when $\lambda = 2$, it becomes Euclidean distance. Chebyshev distance is a special case of Minkowsky distance with $\lambda = \infty$ (taking a limit). This distance can be used for both ordinal and quantitative variables. Formula 70 shows Minkowsky distance for two instances $(\mathbf{x}_r, y_r)$ and $(\mathbf{x}_s, y_s)$ calculated over N features.

$$D_{r,s} = \sqrt[\lambda]{\sum_{j=1}^{N} \left| x_{r,j} - x_{s,j} \right|^{\lambda}} \tag{70}$$

Euclidian distance is the most common and widely used distance function that provides a geometrical distance between two points in a feature space. However, in majority of problem domains features do not represent geometrical distances or their derivatives, and the learning algorithms that use Euclidian distance are not designed to consider this fact. Euclidian distance is not scale-invariant and does not account for correlations between features as

every feature is assumed to be equally important to class discrimination and independent of the others. This assumption may not be always satisfied in real applications, especially in high-dimensional classification problems. In some applications, though, feature selection and dimension reduction is applied to eliminate this problem, while in other applications a preferred choice is a problem-specific distance metric that is capable to identify most important dimensions (Xiang *et al*., 2008).

Manhattan distance, or city-block distance, is an averaged over all dimensions differences between coordinates of two points. It is similar to Euclidian distance, but the particular large differences receive less influence, because the differences are not squared.

The Value Difference Metric (VDM) (Stanfill & Waltz, 1986) was introduced to define an appropriate distance function for nominal (symbolic) features.

$$D_{r,s} = \sum_{j=1}^{N} \left| x_{r,j} - x_{s,j} \right| \tag{71}$$

Bhattacharya distance is based on the probabilistic approach and can be used to evaluate how much each feature contributes to separability of instances from different classes (Duda *et al*., 2001; Devijver & Kittler, 1982). The larger the overlap between the distributions for a certain feature is, the higher uncertainty regarding the class. Therefore, lower Bhattacharya distance corresponds to more "discriminatory" features. Bhattacharya distance is a special case of a more general Chernoff distance.

Many unsupervised and supervised learning algorithms depend upon a good distance function to be successful. In geometrical interpretation, the relative importance of the dimensions in the feature space in development of the distance measurement depends on how the instances are situated in this space, i.e. how much they are stretched or squashed, do they form clusters and how dense they are. Irrelevant and noisy features, as well as redundant features and features with missing values should be excluded from consideration since they will harm representation of the data structure.

For the classification tasks geometrical interpretation of the data in terms of the selected similarity measure or distance function has a goal to represent the instances of different classes in the feature space in the way to provide the most simple and effective class discrimination including class separability. In order to achieve this goal the relevant features should be selected, then the spatial data structure should be put into correspondence with the class labels and finally, class separability may be increased considering feature subspaces or using derived features.

Different distance functions, among which popular choices are Mahalanobis, Bhattacharyya, and Kullback-Leibler, can be used as measures of class separability and estimates of Bayes minimum error in prediction tasks. Manhattan distance, Euclidian distance and Value Difference Metric (VDM) remain among the most popular choices to use in predictive models that involve density estimated and based on the neighborhood concept. It has been

shown that Manhattan distance metric is consistently more preferable than Euclidian distance metric for high-dimensional data mining applications (Aggarval *et al.*, 2001). Following these argumentations, Manhattan distance metric has been used in our implementation of Bidirectional Data Partitioning technique.

Appendix 3

Data sets for validating complexity measures

Synthetic and benchmark data sets used in class separability and complexity measures study are briefly described in this appendix. All descriptions are supplied with graphical projections onto relevant and irrelevant dimensions.

Gauss-2-sep is a two-dimensional data set, has two classes, 1000 instances each, with distributions in class 1 G(5.0, 1.0) – feature 1, and G(10.0, 1.0) – feature 2, and in class 2 G(10.0, 1.0) – feature 1, and G(2.0, 1.0) – feature 2. Gauss-2-sep is an example of linear separability with wide margins between classes. **Gauss-2-one** is a two-dimensional data set, has two classes, 1000 instances each, with distributions in class 1 G(5.0, 1.0) – feature 1, and G(10.0, 1.0) – feature 2, and in class 2 G(10.0, 1.0) – both feature 1 and 2. Gauss-2-one is linearly separable in feature 1, there is a minor intersection between classes, a narrow margin. **Gauss-2-onesep** is a two-dimensional data set, has two classes, 1000 instances each, with distributions in class 1 G(5.0, 0.5) – feature 1, and G(10.0, 0.5) – feature 2, and in class 2 G(10.0, 0.5) – both feature 1 and 2. Gauss-2-onesep is linearly separable in feature 1 with no intersection between classes. **Gauss-2-ov** is a two-dimensional data set, has two classes, 1000 instances each, with distributions G(10.0, 0.5) in both dimensions, both classes, that means classes are completely overlapped. Four of these two-dimensional data sets are shown in Figures 18-21 accordingly.

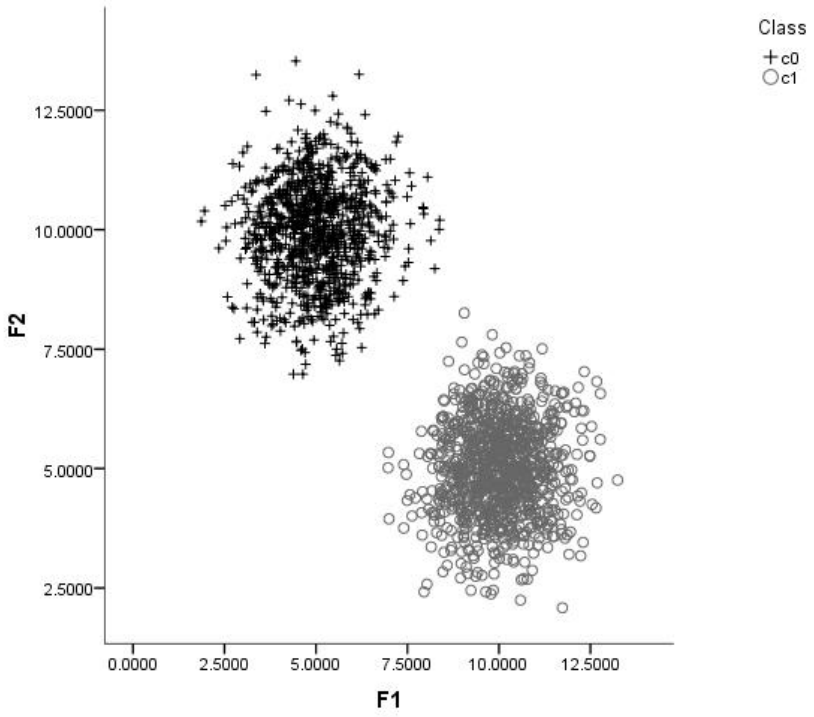

FIGURE 18 Gauss-2-sep two-dimensional data set: both features are discriminative.

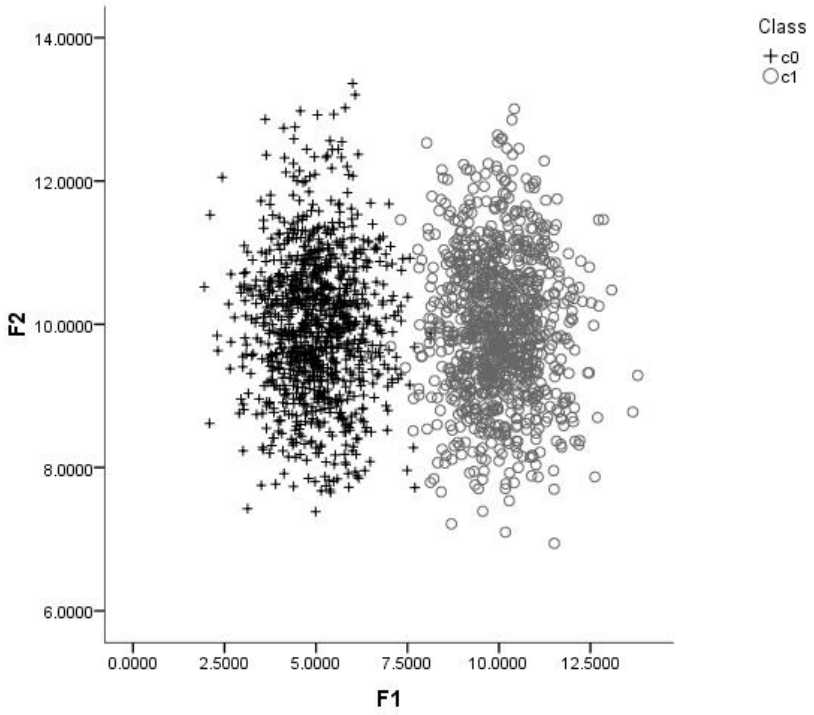

FIGURE 19 Gauss-2-one two-dimensional data set: f_1 is discriminative, narrow margin between classes, f_2 is irrelevant with Gaussian distribution.

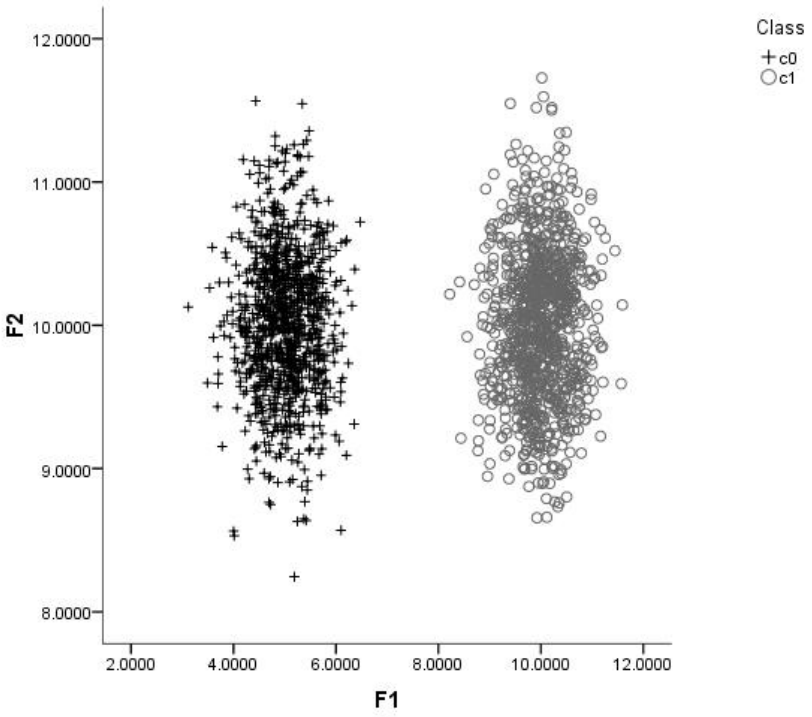

FIGURE 20 Gauss-2-onesep two-dimensional data set: f_1 is discriminative, wide margin between classes, f_2 is irrelevant with Gaussian distribution.

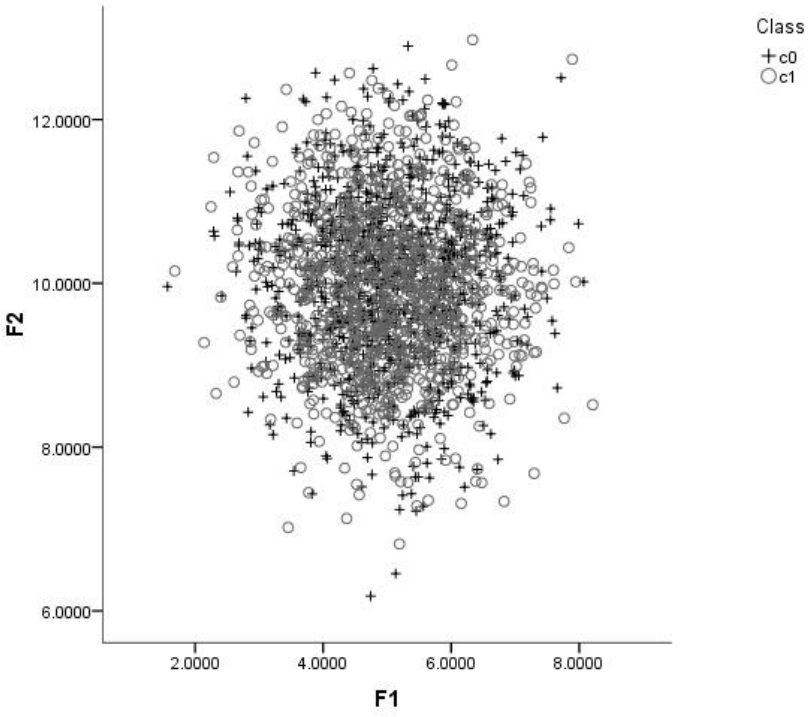

FIGURE 21 Gauss-2-ov two-dimensional data set: both features are irrelevant, Gaussian distribution.

GaussS-2 is a synthetic data set with two Gaussian classes, 2500 instances in each, almost completely separable, with means μA = 5, stdevA = 1 denoted as G(5;1); μB = 10, stdevB = 1 denoted as G(10;1). **GaussS-2+1U** has one additional unimodal irrelevant feature that has a Gaussian distribution G(7.5;3). **GaussS-2+1B** has one additional bimodal irrelevant feature that is a mixture of G(6;3) and G(9;3). **GaussS-2+1M** has one irrelevant feature that is a mixture of G(2;3), G(5;3), and G(8;3). **GaussS-2+1** has one irrelevant feature uniformly distributed in the interval [0…15] denoted as U(0;15). **GaussS-2+all** includes all of the above irrelevant features. Projections onto relevant and irrelevant dimensions are presented below.

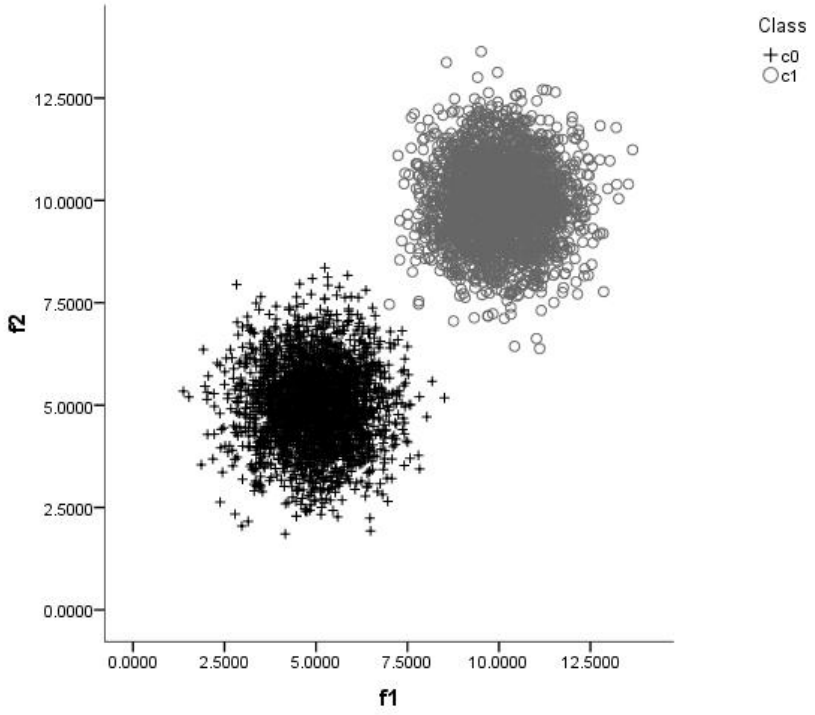

FIGURE 22 GaussS-2 data set shown in two relevant dimensions.

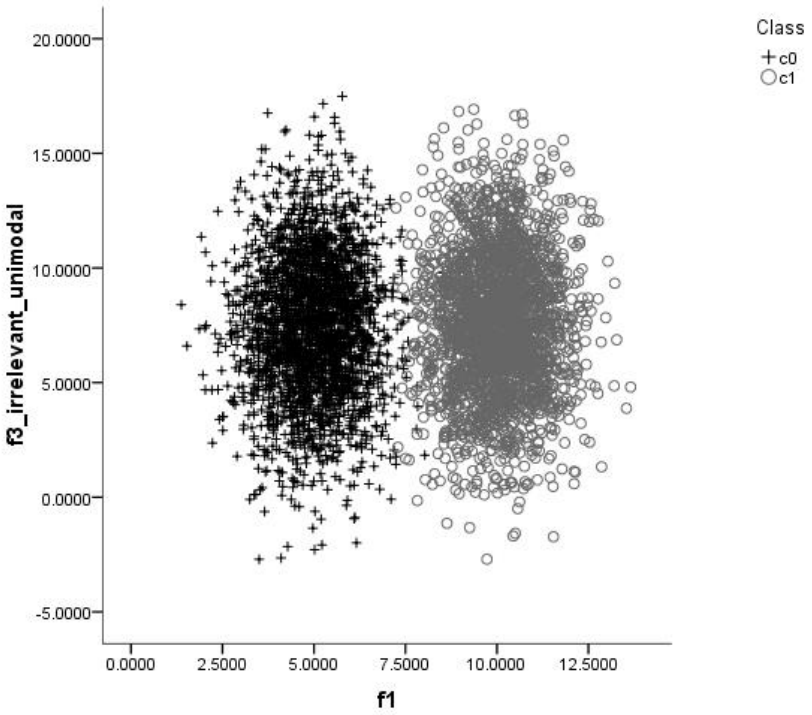

FIGURE 23 GaussS-2+1U data set shown in one relevant dimension and one irrelevant dimension (unimodal Gaussian).

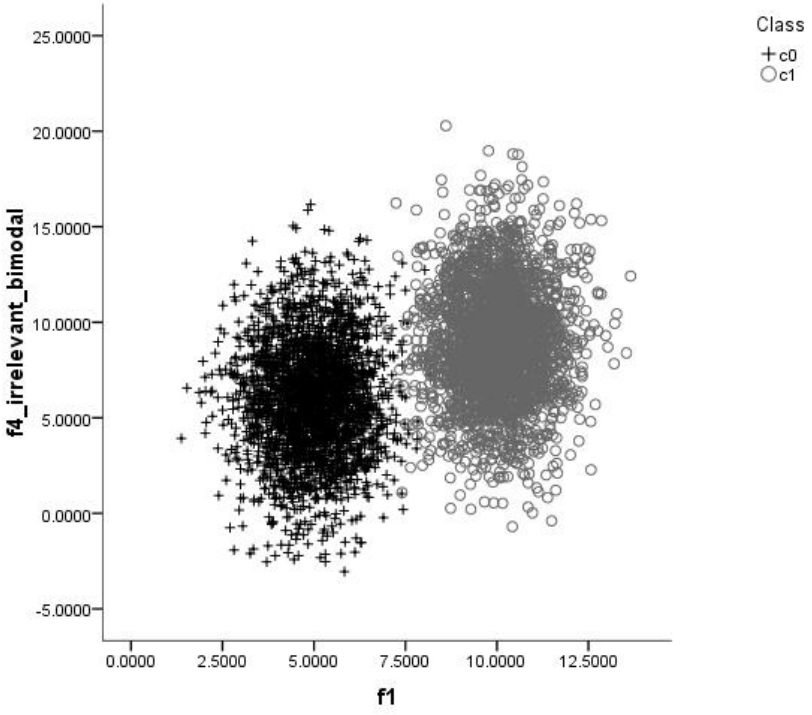

FIGURE 24 GaussS-2+1B data set shown in one relevant dimension and one irrelevant dimension (bimodal Gaussian).

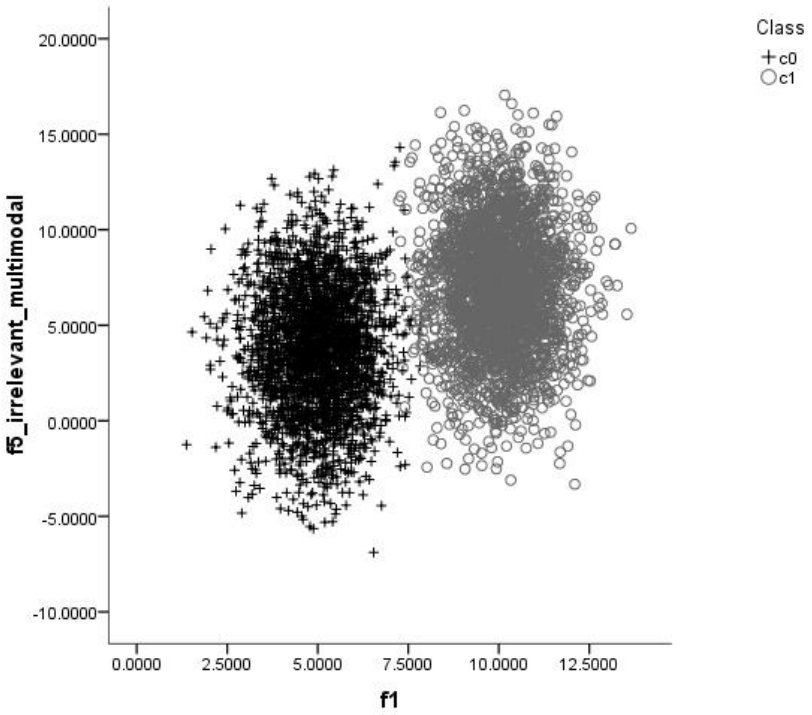

FIGURE 25 GaussS-2+1M data set shown in one relevant dimension and one irrelevant dimension (multimodal Gaussian).

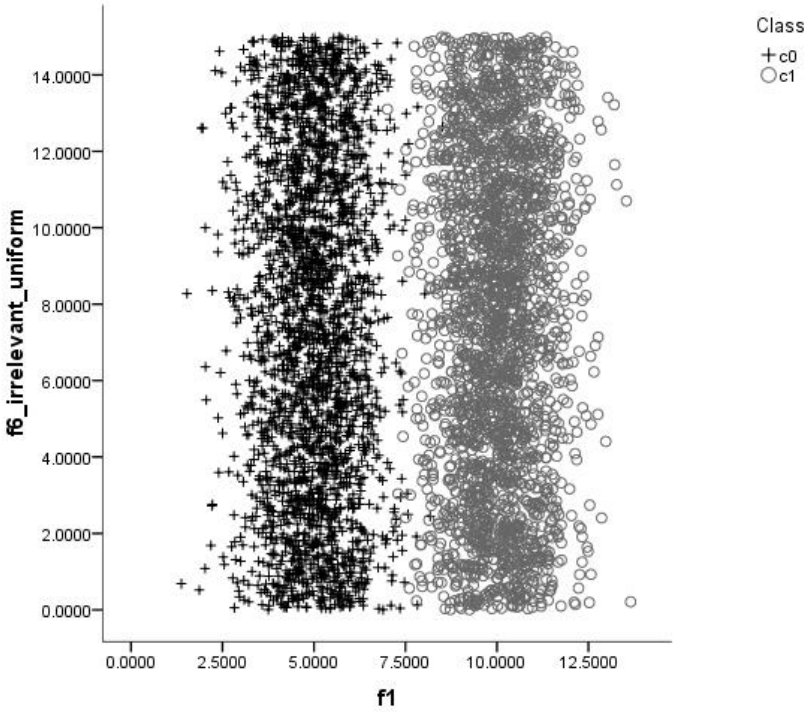

FIGURE 26 GaussS-2+1 data set shown in one relevant dimension and one irrelevant dimension (uniform).

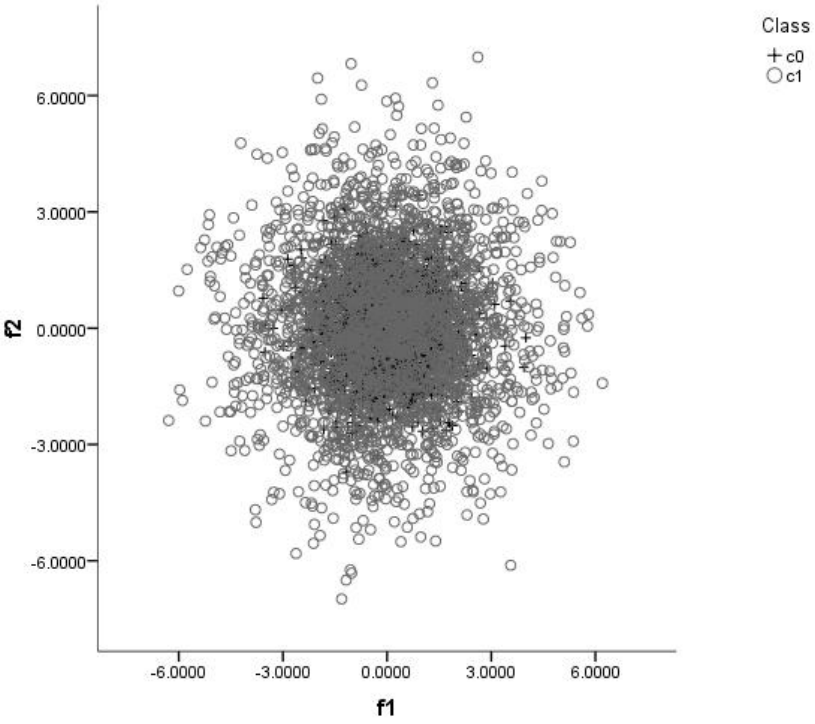

FIGURE 27 Gauss-8+10 data set shown in two relevant dimensions.

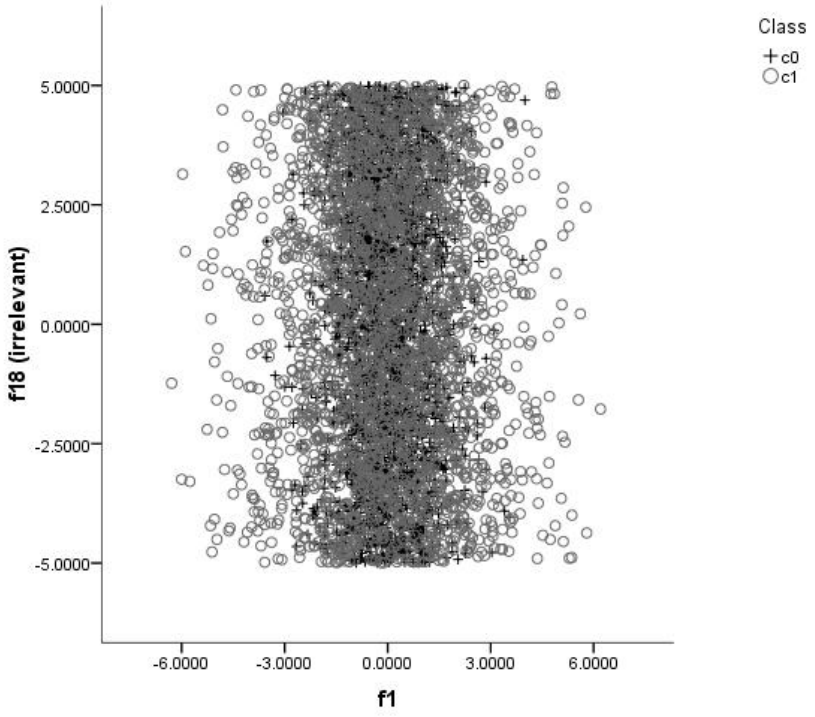

FIGURE 28 Gauss-8+10 data set shown in one relevant and one irrelevant dimension.

Gauss-8 (Blayo *et al.*, 1995) is a data set with 8 continuous features generated according to Gaussian distributions in two classes. The center of gravity is the same for two classes, which makes them fully overlapped. The theoretical error is 9%. **Gauss-8+10** is a modified version of Gauss-8 with 10 irrelevant features, U(-5; 5). It has heavily interleaved classes, approximately equal covariances in classes, equal density in both classes.

FourSubcl-2 data set has two Gaussian subclasses per each of two classes. The first class cl0 is composed of two Gaussian distributions: subclass 1 as G(5.0, 1.0) in both features f1 and f2, and subclass 2 as G(10.0, 1.0) in both features. Two subclasses of class cl1 are f1 - G(10.0, 1.0), f2 G(5.0, 1.0) and f1- G(5.0, 1.0), f2- G(10.0, 1.0). Each subclass is represented by 250 instances in both training and test sets. This data set has a Bhattacharyya upper bound on minimum Bayes error 0.2415. In **FourSubcl-2+5G** there are 5 irrelevant features created with G(7.5, 2.0). In **FourSubcl-2+5U** there are 5 irrelevant features created with U(2.0, 13.0). **FourSubcl-2+10** includes both types of irrelevant features, 5 of each. Graphical representation is shown below.

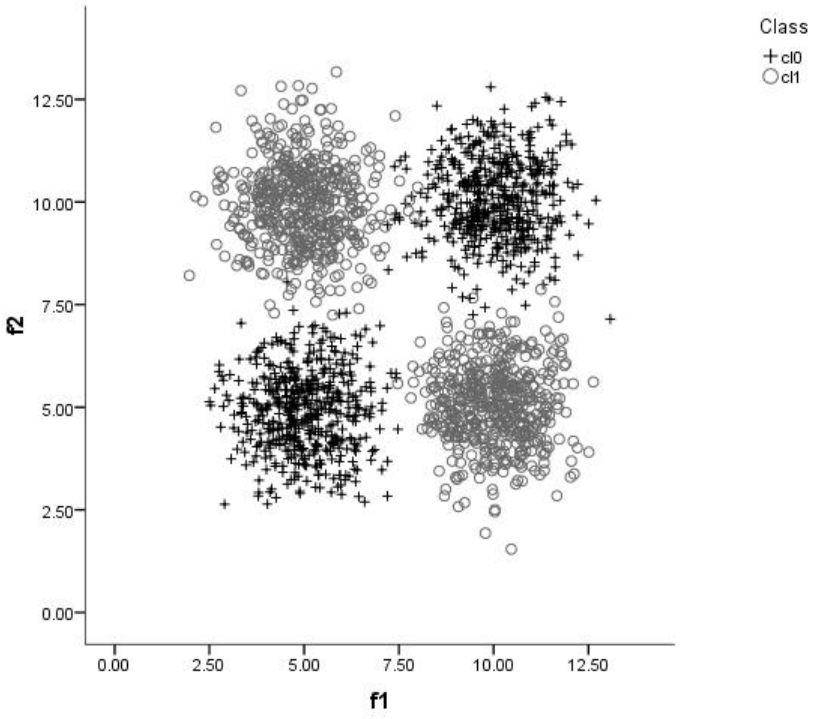

FIGURE 29 FourSubclass-2+10 data set shown in two relevant dimensions.

Clouds-2 data set (Blayo *et al.*, 1995) has 2 classes, one of which has three Gaussian subclasses. One of the subclasses is heavily interleaved with the other Gaussian class, while the other two are partially interleaved. The original data set has 5000 instances, 2500 in each of two classes, 2 continuous numeric features. The theoretical error is 9.66%. **Clouds-2+10** data set has 10 irrelevant features added, U(-3; 3).

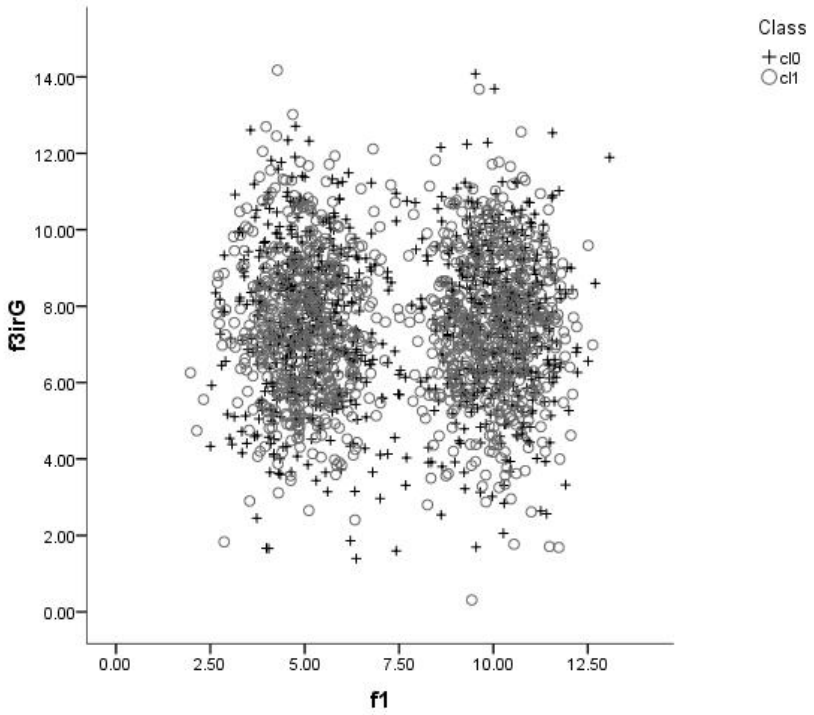

FIGURE 30 FourSubclass-2+10 data set shown in one relevant dimension and one of the Gaussian irrelevant dimensions.

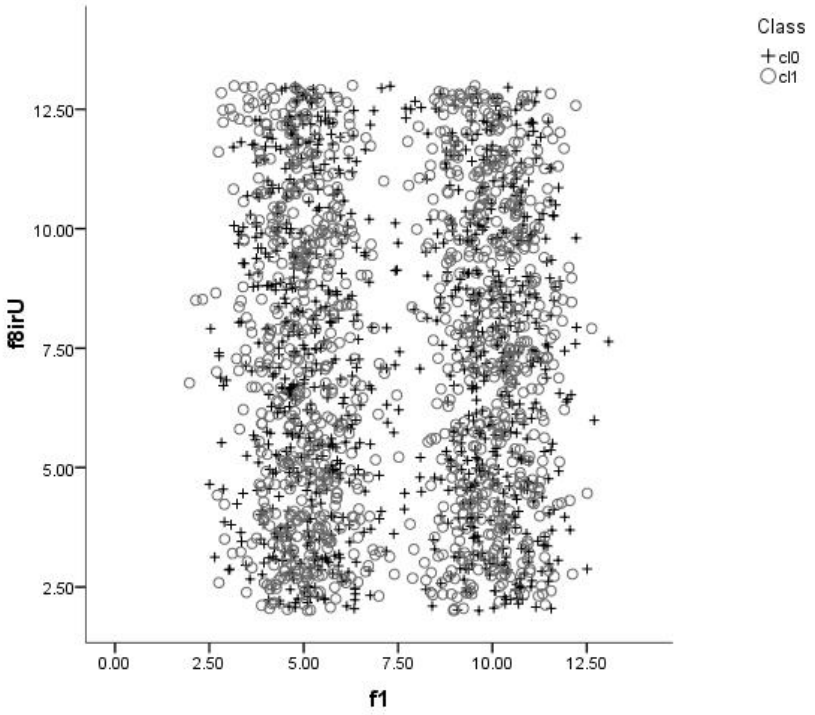

FIGURE 31 FourSubclass-2+10 data set shown in one relevant and one of the irrelevant dimensions with uniform distribution.

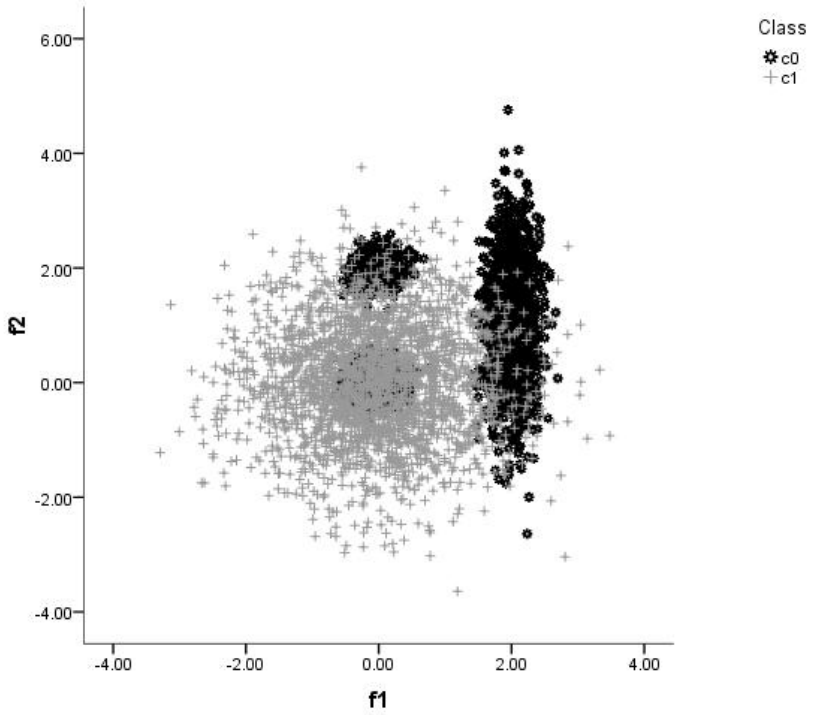

FIGURE 32 Clouds-9 data set shown in one relevant and one irrelevant dimension.

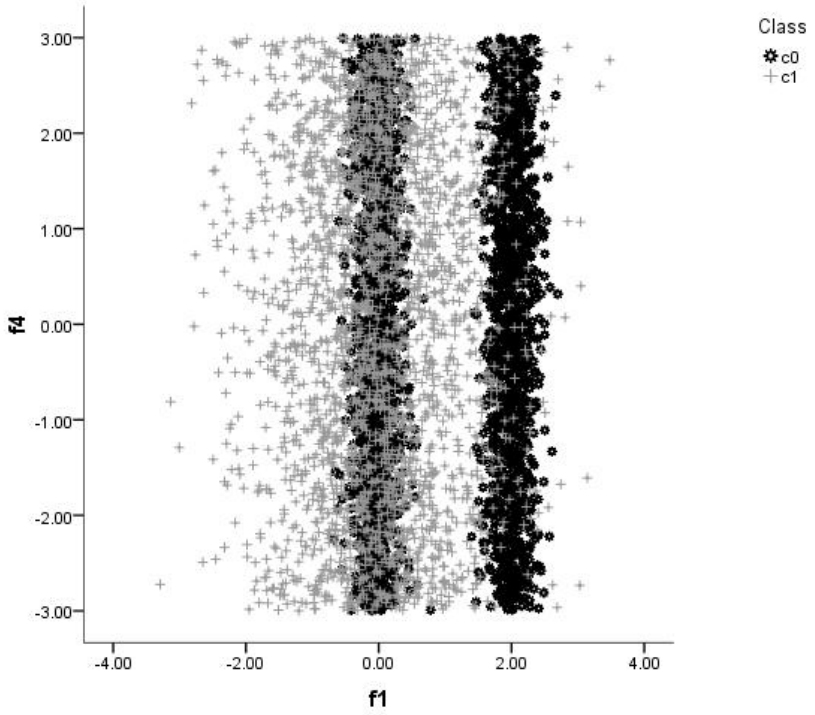

FIGURE 33 Clouds-9 data set: projection onto one relevant and one irrelevant dimension.

Concentric-2 data set (Blayo *et al.*, 1995) consists of two classes: one uniformly distributed within a concentric area, and another class surrounds it without overlapping. This is an example of narrow margin between classes, nonlinear boundary, equal and even density in classes. The original data set has 5000 instances, equal-size classes, and 2 features. The theoretical error is 0%. **Concentric-2+10** has additional 10 irrelevant features, U(-5; 5).

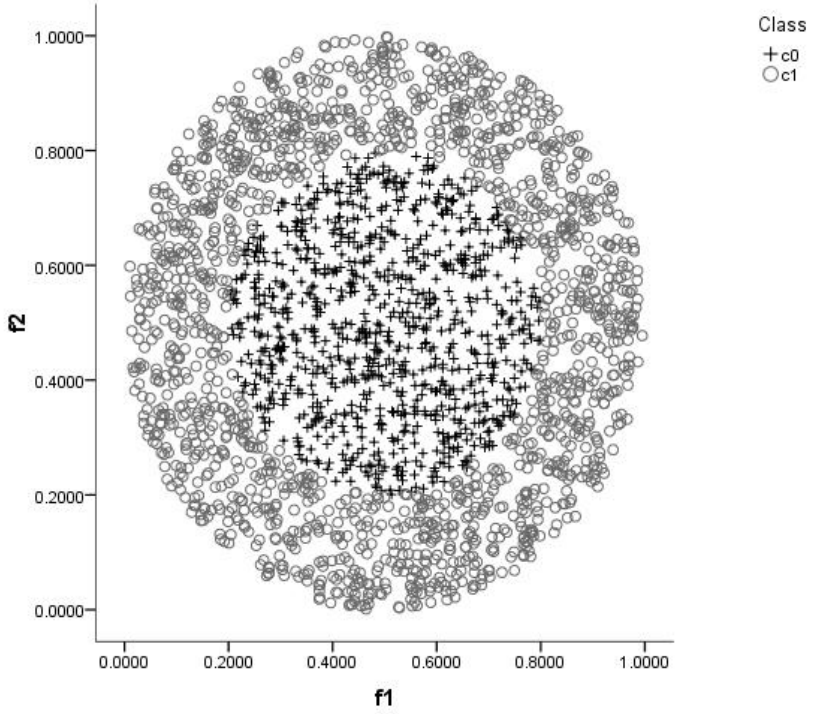

FIGURE 34 Concentric-2+10 data set show in two relevant dimensions.

Spirals-2+3 data set is a modified version of 2-dimensional data presented in (Lindenbaum et al., 1999). It has 3 irrelevant features, uniformly distributed, U(0; 12). In the original **Spirals-2** data set two features take values in the interval (0…20). This data set is an example of nonlinear class boundaries with narrow margins. This data set is presented in Figures II.19 and II.20 below.

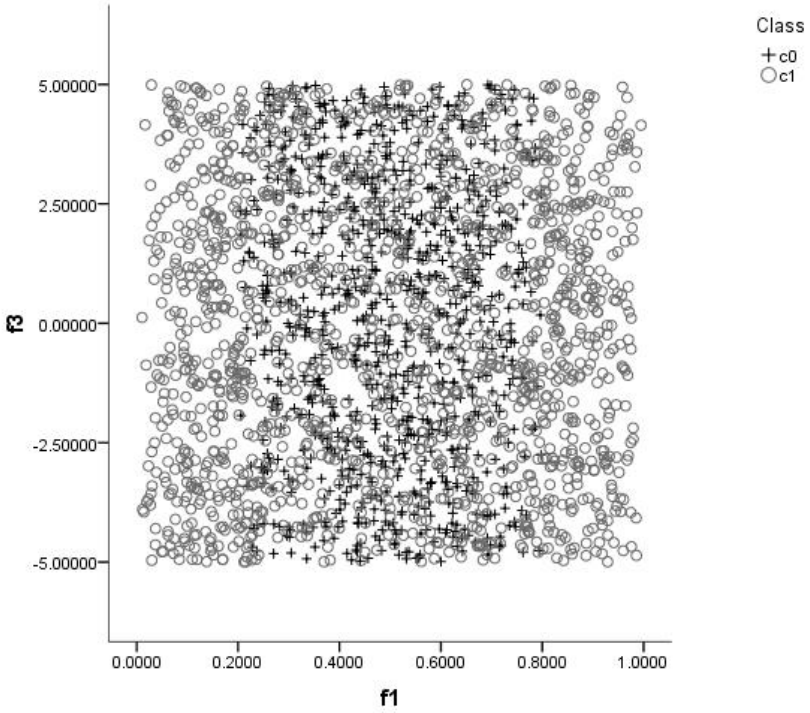

FIGURE 35 Concentric-2+10 data set shown in one relevant and one irrelevant dimension.

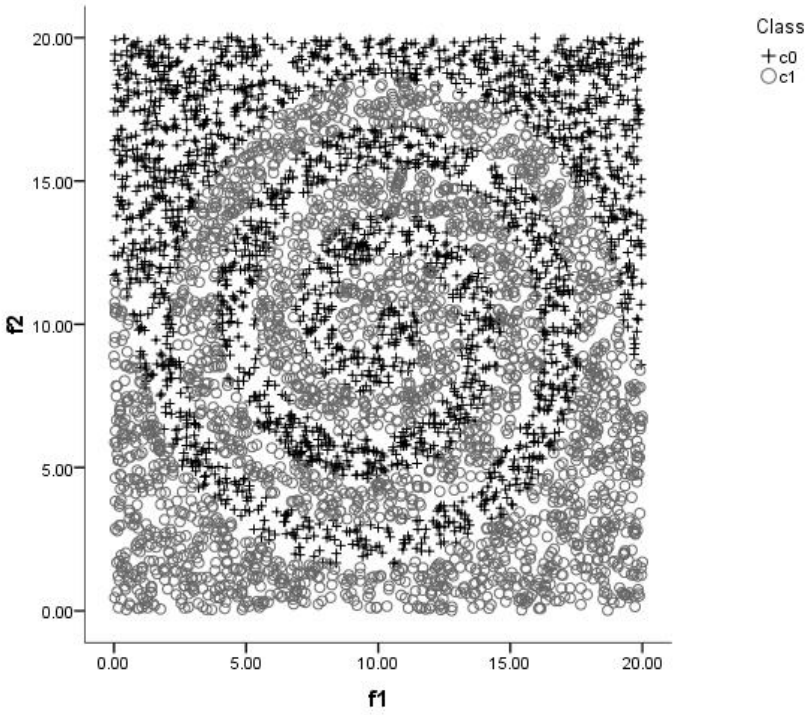

FIGURE 36 Spirals-2+3 data set shown in two relevant dimensions.

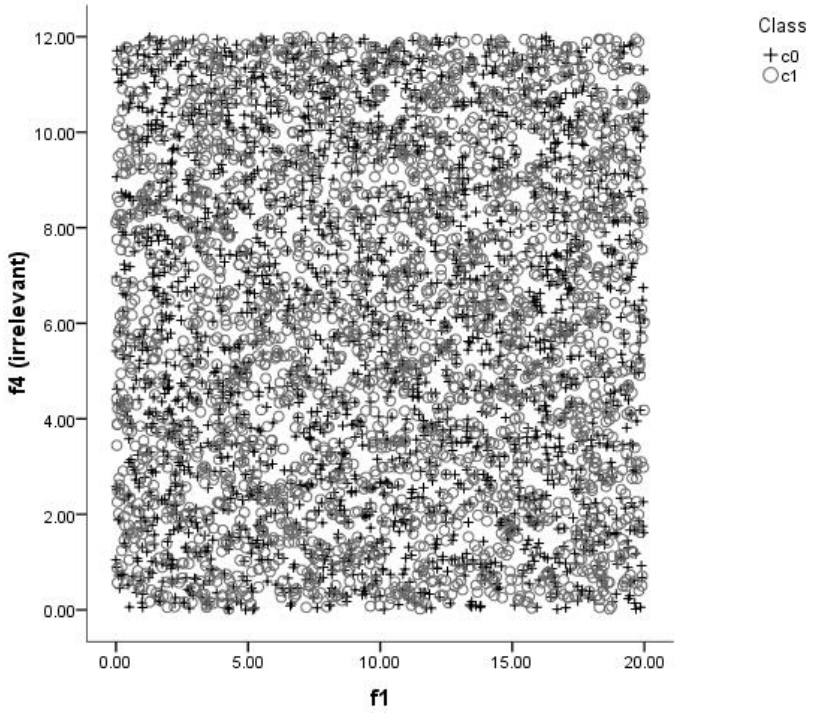

FIGURE 37 Spirals-2+3 data set shown in one relevant dimension and one irrelevant dimension.

Fourclass-2+7M is a data set that presents a case, where irrelevant features obtained from mixed distributions partially capable do discriminate classes, and two relevant features show nonlinear boundary between classes with a wide margin. There are 4 classes (271, 266, 274, and 327 instances), 9 numeric features, 2 relevant and 7 irrelevant features. Fourclass-2 is a data set with two relevant features retained. This data set is a class-balanced 12% sample of the original data set used in (Bernadó-Mansilla & Ho, 2005), which has an equal density unbalanced classes. In **Fourclass-2** classes have unequal density. **Fourclass-2+3** has 3 uniformly distributed irrelevant features, U(0;1). The data set is presented in Figures II.21-II.24.

Birch-2+8 data set is generated using BIRCH data generator in WEKA ("grid" pattern is used). The data set has 3 classes and 10 continuous features, only two of which are partially discriminative. Features #1 and #2 each discriminate a different class from the other two and fully discriminative together; in other features classes are heavily interleaved. There are 11055 instances, classes are balanced: 3695, 3782, 3578 instances respectively. **Birch-2** is the same data set with only two relevant features retained. Graphic presentation is given below, Figures 42 and 43.

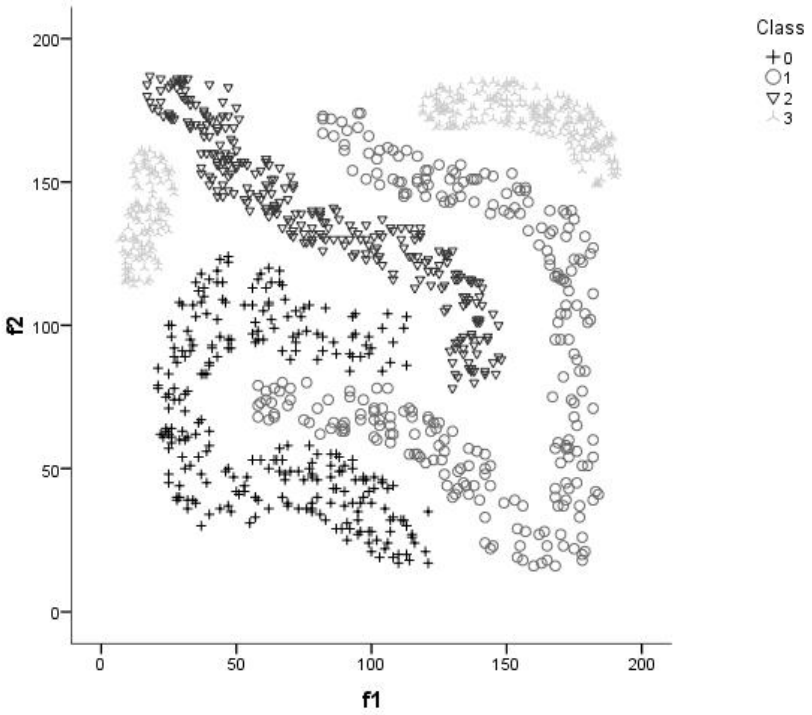

FIGURE 38 Fourclass-2 data set shown in two relevant dimensions.

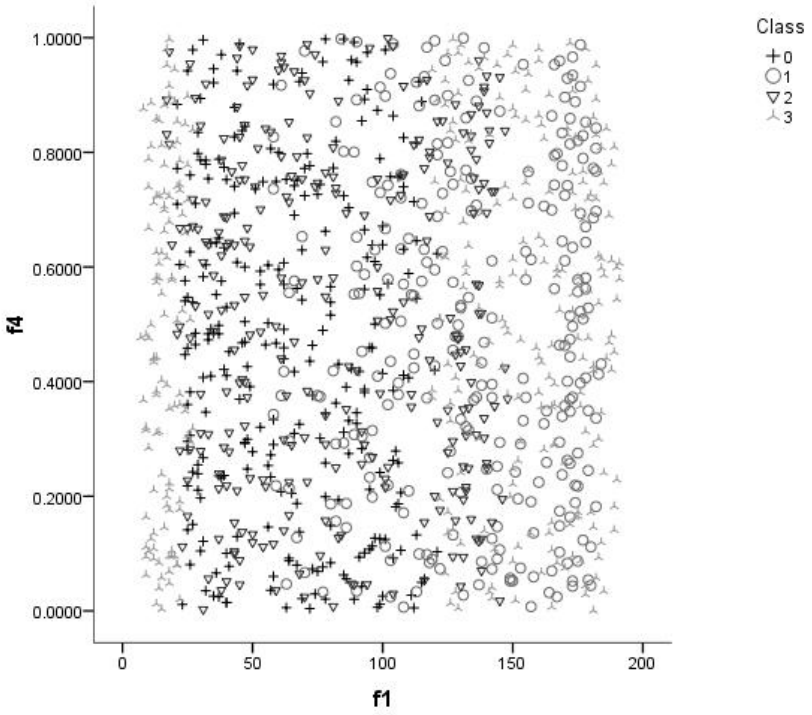

FIGURE 39 Fourclass-2+3 data set shown in one relevant dimension and one irrelevant dimension.

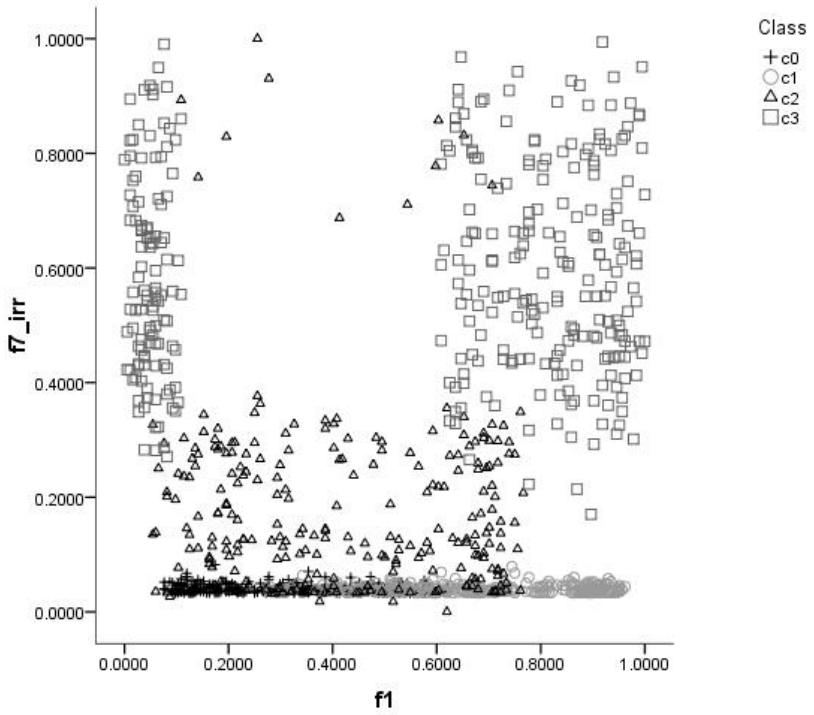

FIGURE 40 Fourclass-2+7M data set shown in one relevant dimension and one irrelevant dimension.

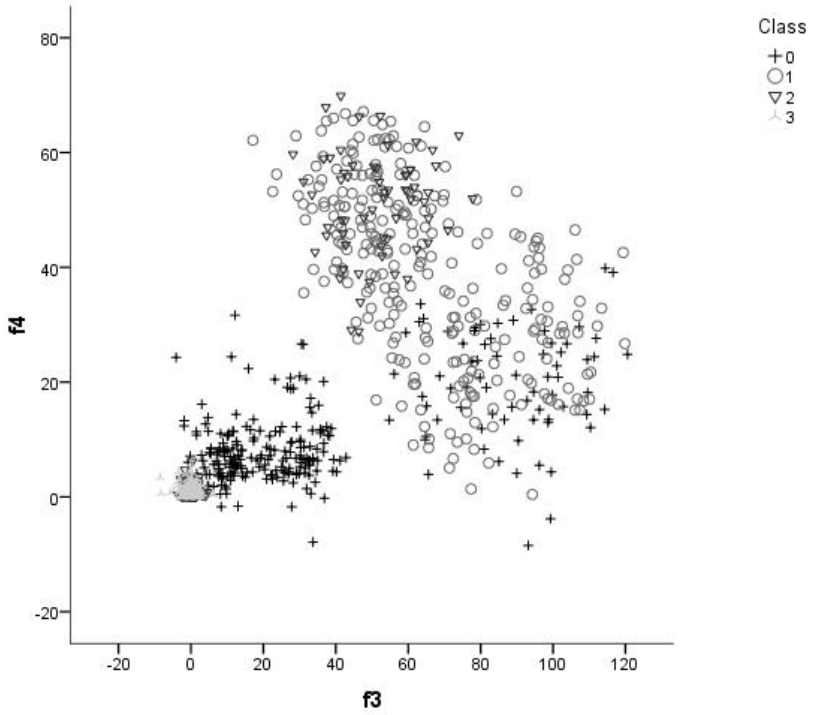

FIGURE 41 Fourclass-2+7M data set show in two irrelevant dimensions.

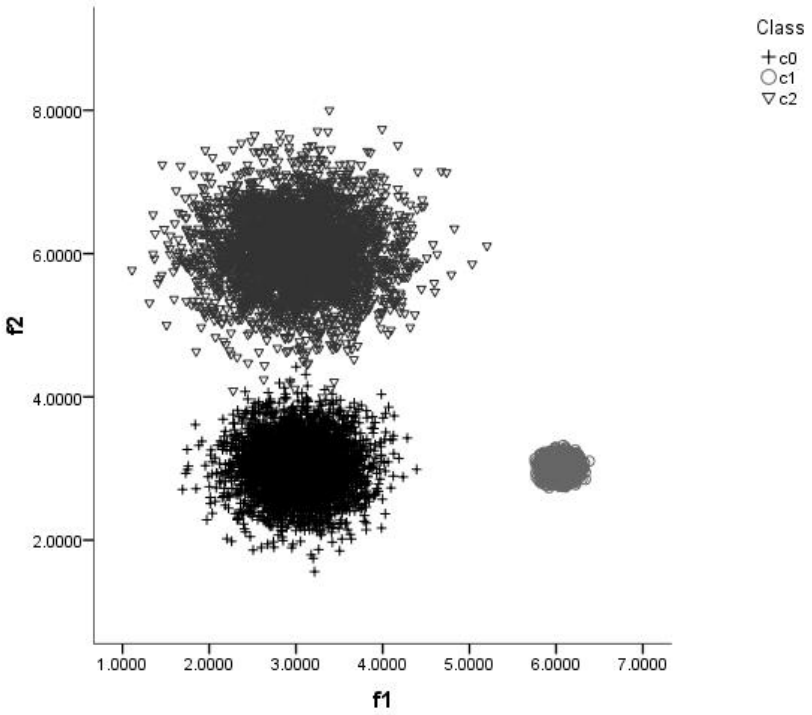

FIGURE 42 Birch-2+8 data set shown in two relevant dimensions.

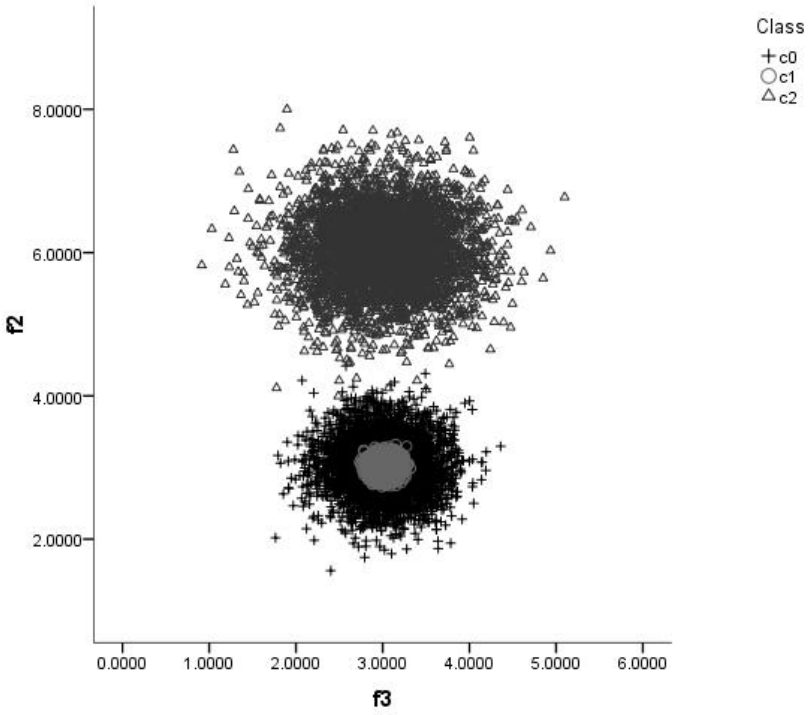

FIGURE 43 Birch-2+8 data set shown in one relevant dimension and one irrelevant dimension.

RBF-10+10 data set is generated in WEKA using RandomRBF generator. The data set is created by first creating a random set of centers for each class following the number of specified centroids. Each center is randomly assigned a weight, a central point per feature, and a standard deviation. To generate new instances, a center is chosen at random taking the weights of each center into consideration. Feature values are randomly generated and offset from the center, where the overall vector has been scaled so that its length equals a value sampled randomly from the Gaussian distribution of the center. The particular center chosen determines the class of the instance. RandomRBF generated data contains only numeric features. In RBF-10+10 there are 5000 instances, 2 classes, 2327 and 2683 instances accordingly, 10 features distributed in the interval (-2...2.5), the number of centroids is 50. In addition, there are 10 irrelevant uniformly distributed features, U(-5; 5). **RBF-10** data set is the same except for these 10 irrelevant features. This data set is an example of Gaussian subclasses in heavily interleaved classes.

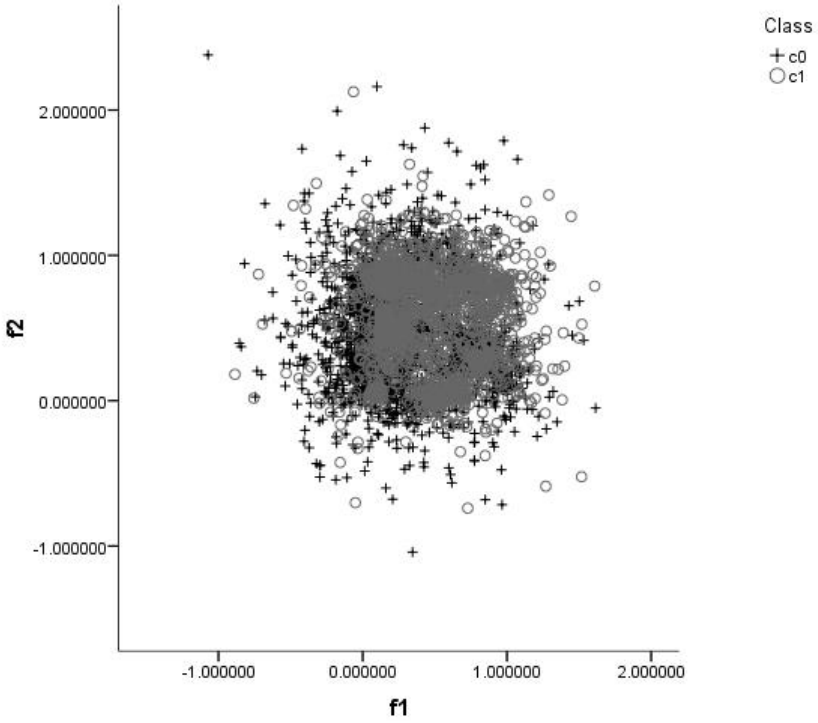

FIGURE 44 RBF-10+10 data set shown in two relevant dimensions.

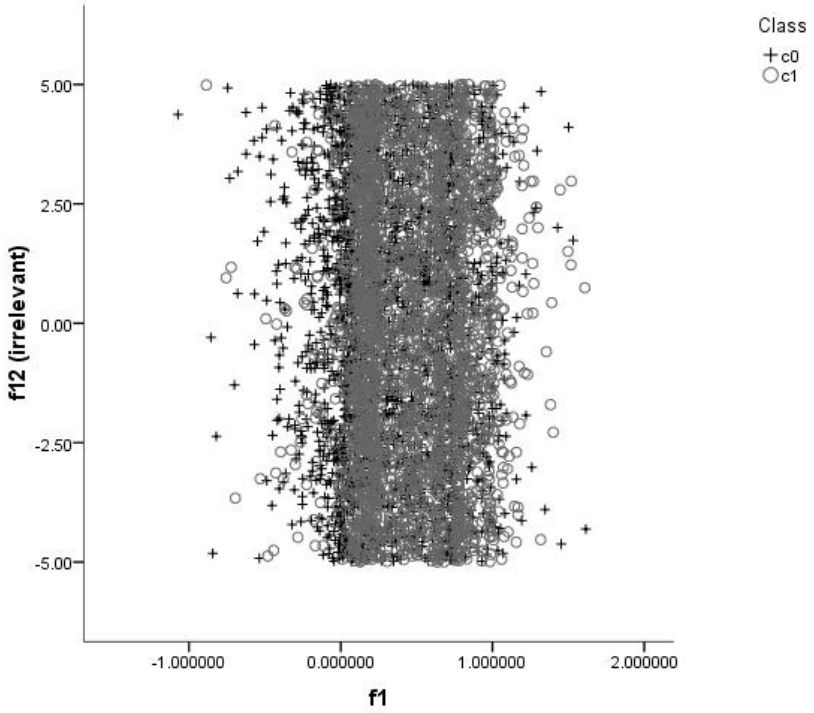

FIGURE 45 RBF-10+10 data set shown in one relevant dimension and one irrelevant dimension.

RDG-10+10 data set have been obtained using WEKA's random data generator RDG1. It creates data randomly by producing a decision list consisting of rules. Instances are generated randomly one by one. If the decision list fails to classify the current instance, a new rule according to this current instance is generated and added to the decision list. RDG-10+10 has 10 continuous relevant features distributed in the interval (-1.5…2.5) and additional 10 irrelevant features, U(-5; 5). The maximum and minimum numbers of tests in rules are set to 10 and 1 accordingly. There are 2 classes, 2548 and 2452 instances. RDG-10 data set is the same except for the 10 irrelevant features. Data more elongated in irrelevant dimensions compared to relevant dimensions, but there is no visual distinction between relevant and irrelevant features in terms of class boundaries. Classes are heavily interleaved.

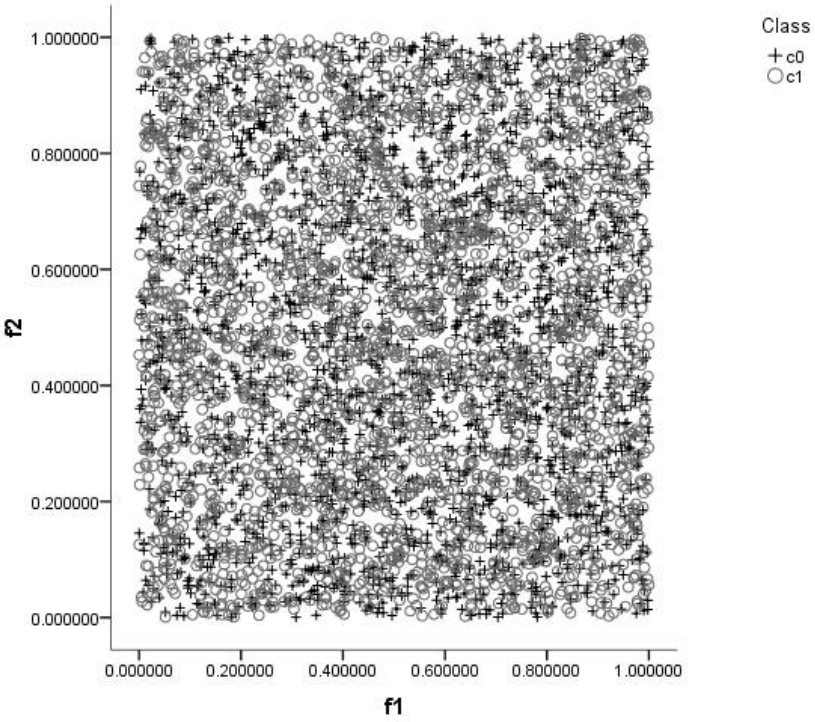

FIGURE 46 RDG-10+10 data set shown in two relevant dimensions.

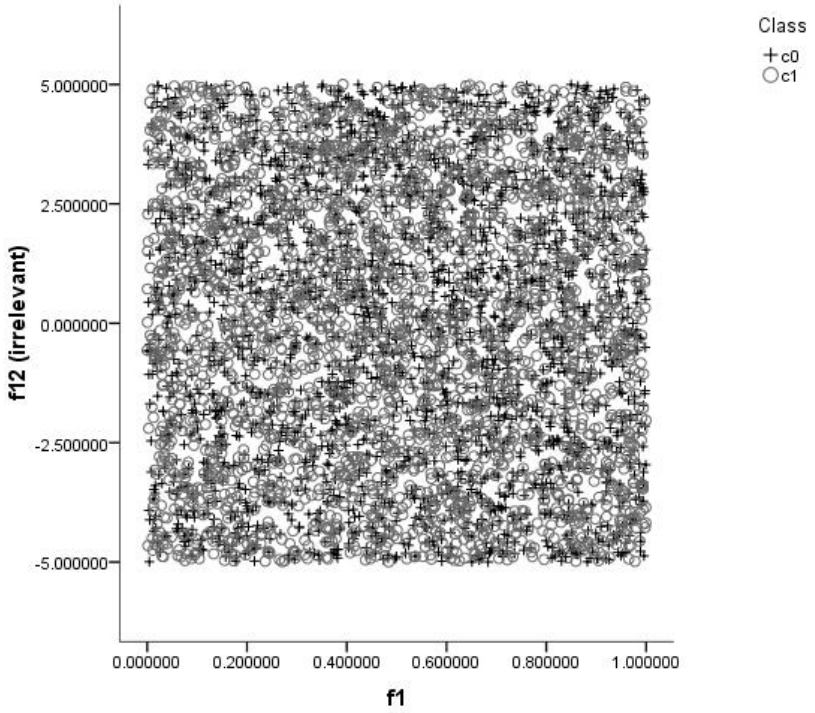

FIGURE 47 RDG-10+10 data set shown in one relevant dimension and one irrelevant dimension.

Appendix 4

Author's additions to WEKA software package

The novel Bidirectional Data Partitioning (BDP) technique has been implemented using WEKA open source data mining package, the latest stable release WEKA 3.6.6. WEKA (Hall et al., 2009) is a collection of machine learning algorithms for solving real-world data mining problems. WEKA API can be found at [http://weka.sourceforge.net/doc.stable/].

In order to support BDP implementation, author has extended functionality of WEKA with addition of the following classes:

- weka.classifiers.meta.BidirectionalPartitioning;
- weka.clusterers.DBScanWeighted;
- weka.clusterers.KMeansWeighted;
- weka.clusterers.HierarchicalClustererWeighted;
- weka.clusterers.forOPTICSAndDBSCAN.Databases.DatabaseWeighted;
- weka.clusterers.forOPTICSAndDBSCAN.Databases.SequentialDatabaseWeighted;
- weka.clusterers.forOPTICSAndDBSCAN.DataObjects.DataObjectWeighted;
- weka.clusterers.forOPTICSAndDBSCAN.DataObjects.ManhattanDataObjectWeighted;
- weka.core.InstancesWeighted;
- weka.core.InstanceWeighted;
- weka.filters.supervised.instance.RemoveInconsistent.

Additional analysis goals, such as preliminary heterogeneity tests, have motivated author to create a weka.analysers package and add the following classes:

- weka.analysers.Analyser;

- weka.analysers.AnalyserEvaluation;
- weka.analysers.AdherenceMapping;
- weka.analysers.IPABasic;
- weka.analysers.IPAEvaluation;
- weka.analysers.MultiClassAnalyser;
- weka.analysers.WeightedDistanceAnalyser;
- weka.analysers.UpdatableAnalyser.

This package is currently under development and will be supplied with class separability and complexity tests which are currently a part of BDP. Another addition planned is a sampling technique with a stochastic component that will serve as wrapper for heterogeneity tests. ContextualPartitioning technique and its component, R-IPA tree-like search procedure, are currently under development as well.

In order to handle higher order feature dependencies, author has developed a filter that merges values of two features with nominal or discrete numeric values:

- weka.filters.unsupervised.MergeTwoValues,
- weka.filters.unsupervised.attribute.

Stochastic Discrimination technique used in supporting studies on coverage optimization ensemble techniques has been included among WEKA's meta-classifiers:

- weka.classifiers.meta.sd.StochasticDiscrimination.

This implementation uses the original variant with random subspaces and author's multi stream version with neighborhood search (Skrypnyk, 2009; Skrypnyk & Ho, 2006).

Screenshots demonstrating BDP within WEKA, parameter settings and BDP results, are shown in Figures VI.1 and VI.2 correspondingly.

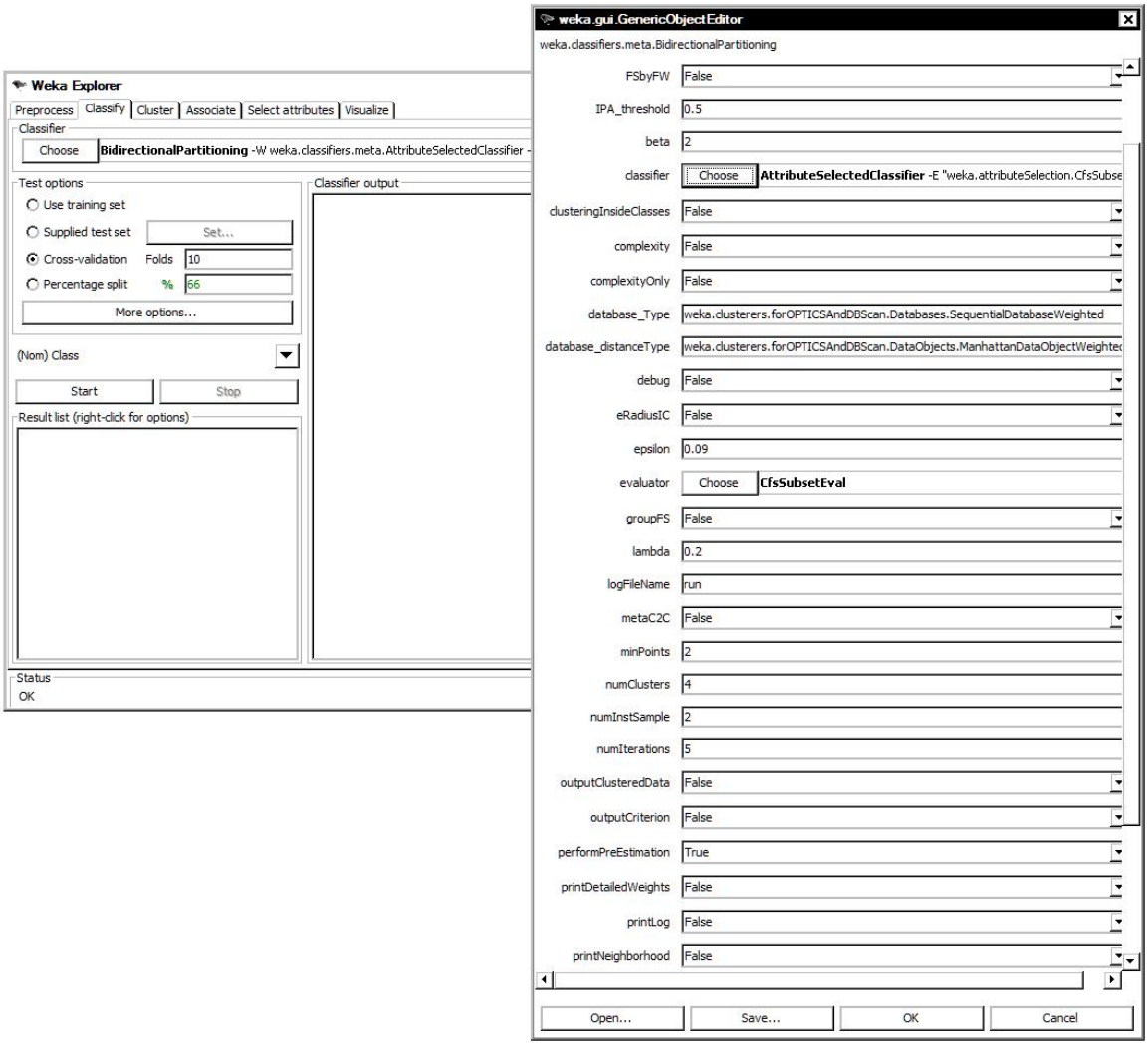

FIGURE 48 BidirectionalPartitioning in WEKA, parameter setting.

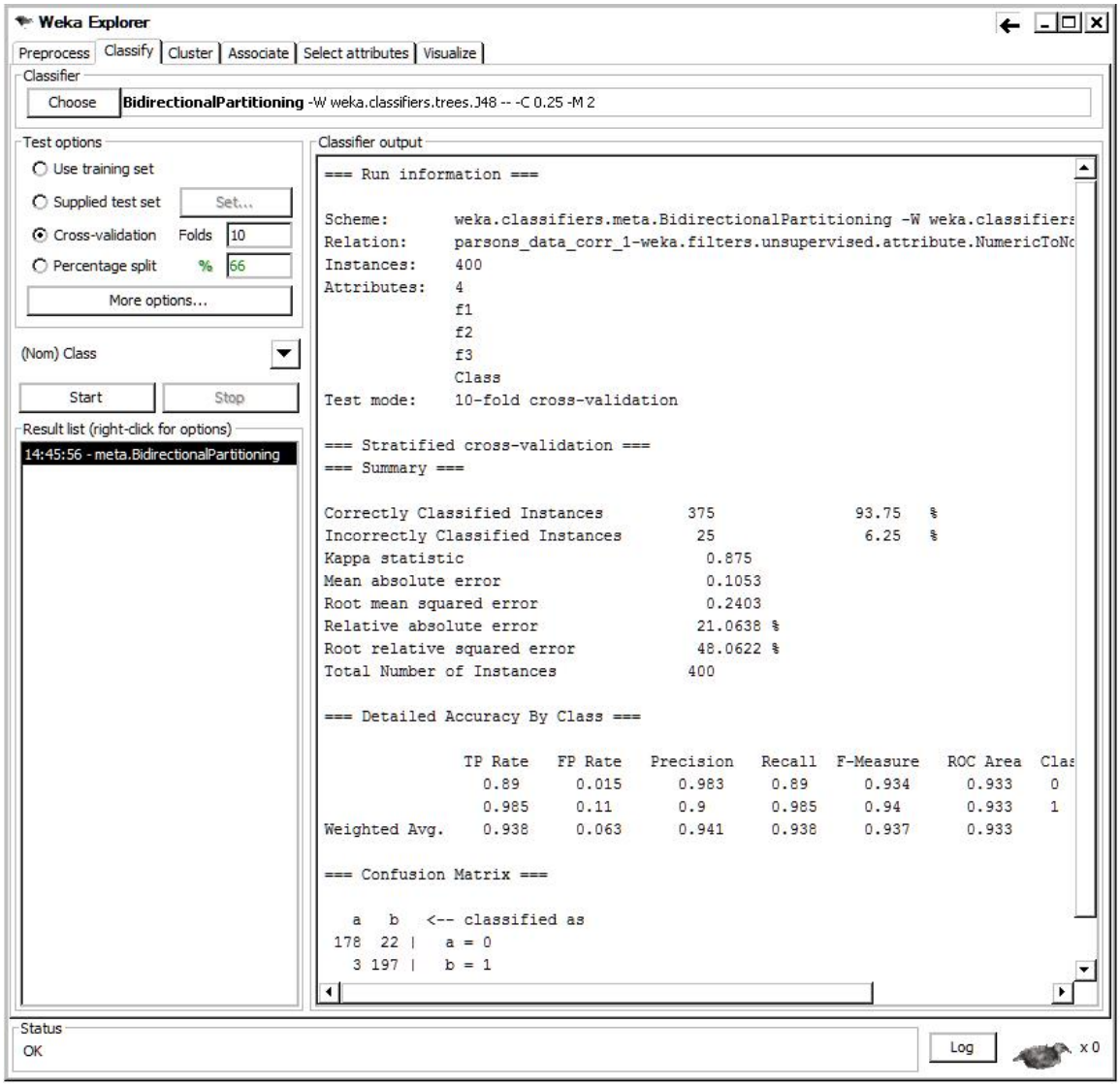

FIGURE 49 BidirectionalPartitioning in WEKA, classification results.

Decomposition approach developed for class heterogeneity is currently implemented in WEKA and available through combination of the following:

- weka.classifiers.meta.MultiClassClassifier;
- weka.classifiers.meta.AttributeSelectedClassifier.

Appendix 5

Genomics basics

Material presented in this appendix have been acquired from various electronic publications from leading research centers and scientific news media sources (Piatetsky-Shapiro & Tamayo, 2003; Cancer Genome, 2011; National Institute of General Medical Sciences, 2011; Genetics Home Reference, 2011; National Human Genome Research Institute, 2011; Gene Expression Mechanism, 2011; Cancer and Cancer Genetics, 2011).

5.1 Basic notions from bioinformatics

DNA, or deoxyribonucleic acid, carries the hereditary material in humans' and almost all other organisms' cells. Nearly every cell in a person's body has the same DNA. Most DNA is located in the cell nucleus (where it is called nuclear DNA), but a small amount of DNA can also be found in the mitochondria (where it is called mitochondrial DNA or mtDNA). Researchers refer to DNA found in the cell's nucleus as nuclear DNA. An organism's complete set of nuclear DNA is called its *genome*.

DNA is made up of four similar chemicals (called bases and abbreviated A (adenine), T (thymine), C (cytosine), and G (guanine)) that are repeated over and over in pairs. Human DNA consists of about 3 billion bases, and more than 99 percent of those bases are the same in all people. The order, or sequence, of these bases determines the information available for building and maintaining

an organism, similar to the way in which letters of the alphabet appear in a certain order to form words and sentences.

DNA bases pair up with each other, A with T and C with G, to form units called base pairs. Each base is also attached to a sugar molecule and a phosphate molecule. Together, a base, sugar, and phosphate are called a *nucleotide*. Nucleotide is the structural unit of nucleotide chains forming nucleic acids as RNA and DNA. Nucleotides are arranged in two long strands that form a spiral called a double helix. In this double helix, the sugar and phosphate molecules form the vertical sidepieces, and the base pairs form the connecting rungs like in a twisted ladder.

An important property of DNA is that it can replicate, or make copies of itself. Each strand of DNA in the double helix can serve as a pattern for duplicating the sequence of bases. This is critical when cells divide because each new cell needs to have an exact copy of the DNA present in the old cell.

DNA is found inside a special area of the cell called the *nucleus*. Because the cell is very small, and because organisms have many DNA molecules per cell, each DNA molecule must be tightly packaged. This packaged form of the DNA is called a *chromosome*.

DNA spends a lot of time in its chromosome form. But during cell division, DNA unwinds so it can be copied and the copies transferred to new cells. DNA also unwinds so that its instructions can be used to make proteins and for other biological processes.

Genes, being made of DNA, serve as coded instructions for making functional molecules such as ribonucleic acid (RNA) and proteins, which perform the chemical reactions in human bodies. A gene is a distinct portion of a cell's DNA. Human beings have about 25,000 genes. Researchers have discovered functions for some of human genes, and have identified those associated with disorders (such as cystic fibrosis or Huntington's disease). There are, though, many genes whose functions are still unknown.

The human genome is a complete copy of the entire set of human gene instructions. The Human Genome Project, completed in 2003, identified all the human genes in DNA and stored the information in databases so all researchers everywhere could use it.

The particular order of the DNA bases pairs is extremely important in the DNA. Sometimes a replication mistake occurs and one of the pairs gets switched, dropped, or repeated. This changes the coding for one or more genes. This is called genetic mutation. A mutation maybe disease-causing or harmless.

Another way the DNA code could be changed is by errors in the chromosomes. Parts of a chromosome could break off, switch with part of another chromosome, or be swapped within the same chromosome. If any of these or other mistakes occurs then changes (mutations) happen in the gene coding. Sometimes there may be 3 or more copies of a chromosome, or only one chromosome, instead of the normal pair.

Proteins are chains of chemical building blocks called amino acids. A protein could contain just a few amino acids in its chain or it could have several

thousands. Proteins form the basis for most of what the body does, such as digestion, making energy and growing.

5.2 DNA microarray techniques

DNA microarray techniques measure the expression level of thousands of genes in a single experiment. Gene expression is the process by which information from a gene is used in the synthesis of a functional gene product. For protein-coding genes this functional product is protein, for non-protein coding genes such as rRNA genes or tRNA genes, the product is a functional RNA. A DNA microarray (also called gene chip or biochip) is a collection of microscopic DNA spots attached to a solid surface. Each DNA spot contains picomoles of a specific DNA sequence, known as probes (or reporters), which can be a short section of a gene or other DNA element. In gene expression microarray data each instance is presented by thousands of genes as features.

The procedure of obtaining histopathological information for cancer diagnostics is following. The tissue is removed, placed in a fixative to prevent decay, and then prepared using histology procedures for viewing under a microscope. After the tissue processing paraffin will replace the water in the tissue, turning soft, moist tissues into a sample miscible with paraffin, so the sample can be cut into very thin sections. The slices are thinner than the average cell, and are layered on a glass slide for staining. To see the tissue under a microscope, the sections are stained with one or more pigments. The aim of staining is to reveal cellular components. Counterstains are used to provide contrast. For example, antibodies are used to stain specific proteins, lipids and carbohydrates. This technique allows to specifically identify categories of cells under a microscope. Other advanced techniques include in situ hybridization to identify specific DNA or RNA molecules. Digital cameras are increasingly used to capture histopathological images.

Gene sequencing refers to the process of recording the exact sequence of nucleotides in the section of an organism's DNA corresponding to a specific gene. The complete genetic sequences of humans and many other organisms have been determined. Researchers sometimes sequence specific genes of an individual with a certain phenotype (such as a disease) in an attempt to discover the phenotype's genetic basis.

A DNA microarray is a collection of microscopic DNA spots attached to a solid surface, such as glass, plastic or silicon chip forming an array. The affixed DNA segments are known as *probes*, thousands of which can be used in a single DNA microarray. DNA microarrays are used to measure the expression levels of large numbers of genes simultaneously. This is relevant to many areas of biology and medicine, such as studying treatments, disease and developmental stages. Gene expression (also protein expression, or often simply expression) is the process by which a gene's information is converted into the structures and functions of a cell.

Each data point produced by a DNA microarray hybridization experiment represents the ratio of expression levels of a particular gene under two different

experimental conditions. The result, from an experiment with n genes on a single chip, is a series of n expression-level ratios. Typically, the numerator of each ratio is the expression level of the gene in the varying condition of interest, whereas the denominator is the expression level of the gene in some reference condition. The data from a series of m such experiments may be represented as a gene expression matrix, in which each of the n rows consists of an m-element expression vector for a single gene. The expression measurement is positive if the gene is induced (turned up) with respect to the reference state and negative if it is repressed (turned down).