Université d'Ottawa • University of Ottawa

PERMISSION DE REPRODUIRE ET DE DISTRIBUER LA THÈSE

PERMISSION TO REPRODUCE AND DISTRIBUTE THE THESIS

NOM DE L'AUTEUR / NAME OF AUTHOR:	Sylvain Létourneau
ADRESSE POSTALE / MAILING ADDRESS:	224 du Voilier Gatineau, Quebec J8P 7Z4
GRADE / DEGREE:	**ANNÉE D'OBTENTION / YEAR GRANTED**
Ph.D - (Computer Science)	2003
TITRE DE LA THÈSE / TITLE OF THESIS:	
Identification of Attribute Interactions and Generation of Globally Relevant Continuous Features in Machine Learning	

L'auteur permet, par la présente, la consultation et le prêt de cette thèse en conformité avec les règlements établis par le bibliothécaire en chef de l'Université d'Ottawa. L'auteur autorise aussi l'Université d'Ottawa, ses successeurs et cessionnaires, à reproduire cet exemplaire par photograhie ou photocopie pour fins de prêt ou de vente au prix coûtant aux bibliothèques ou aux chercheurs qui en feront la demande.

Les droits de publication par tout autre moyen et pour vente au public demeureront la propriété de l'auteur de la thèse sous réserve des règlements de l'Université d'Ottawa en matière de publication de thèses.

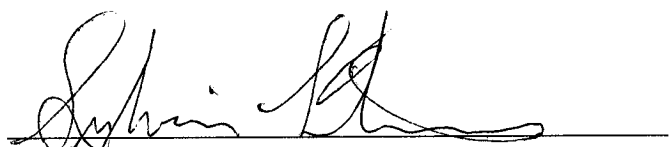
The author hereby permits the consultation and the lending of this thesis pursuant to the regulations established by the Chief Librarian of the University of Ottawa. The author also authorizes the University of Ottawa, its successors and assignees, to make reproductions of this copy by photographic means or by photocopying and to lend or sell such reproductions at cost to libraries and to scholars requesting them.

The right to publish the thesis by other means and to sell it to the public is reserved to the author, subject to the regulations of the University of Ottawa governing the publication of theses.

N.B. LE MASCULIN COMPREND ÉGALEMENT LE FÉMININ

2003-09-08

DATE



(AUTEUR) SIGNATURE (AUTHOR)

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

LÉTOURNEAU, Sylvain

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

Ph. D. (Computer Science)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

Identification of Attribute Interactions and Generation of Globally Relevant Continuous Features in Machine Learning

S. Matwin

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

F. Famili

CO-DIRECTEUR DE LA THÈSE - THESIS CO-SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

R. Caruana

R. Holte

N. Japkowicz

J. Oommen

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

Identification of Attribute Interactions and Generation of Globally Relevant Continuous Features in Machine Learning

Sylvain Létourneau

A dissertation submitted to the School of Graduate Studies
and Research of the University of Ottawa in partial fulfillment
of the requirements for the degree of Doctor of Philosophy

Ottawa-Carleton Institute for Computer Science

School of Information Technology and Engineering (SITE)
University of Ottawa
Ottawa, Ontario, Canada

Contents

List of Tables

List of Figures

Abstract

Datasets found in real world applications of machine learning are often characterized by low-level attributes with important interactions among them. Such interactions may increase the complexity of the learning task by limiting the usefulness of the attributes to dispersed regions of the representation space. In such cases, we say that the attributes are *locally* relevant. To obtain adequate performance with locally relevant attributes, the learning algorithm must be able to analyse the interacting attributes simultaneously and fit an appropriate model for the type of interactions observed. This is a complex task that surpasses the ability of most existing machine learning systems. This research proposes a solution to this problem by extending the initial representation with new *globally* relevant features. The new features make explicit the important information that was previously hidden by the initial interactions, thus reducing the complexity of the learning task.

This dissertation first proposes an idealized study of the potential benefits of globally relevant features assuming perfect knowledge of the interactions among the initial attributes. This study involves synthetic data and a variety of machine learning systems. Recognizing that not all interactions produce a negative effect on performance, the dissertation introduces a novel technique named *Relevance graphs* to identify the interactions that negatively affect the performance of existing learning systems. The tool of *interactive relevance graphs* addresses another important need by providing the user with an opportunity to participate in the construction of a new representation that cancels the effects of the negative attribute interactions. The dissertation extends the concept of relevance graphs by introducing a series of algorithms for the automatic discovery of appropriate transformations. We use the named GLOREF (GLObally Relevant Features) to designate the approach that integrates these algorithms. The dissertation fully describes the GLOREF approach along with an extensive empirical evaluation with both synthetic and UCI datasets. This evaluation shows that the features produced by the GLOREF approach significantly improve the accuracy with both synthetic and real-world data.

Acknowledgments

Over the length of this work, I remember having said *thank you* to many people for their help and support. It is now with great pleasure that I use this opportunity to write down my gratitude to these people.

First, I would like to thanks my supervisor Prof Stan Matwin for his unconditional support. Stan deployed extraordinary efforts to provide me with a proper environment for my research. Even before completing the first term of the PhD program, Stan introduced me to the NRC (National Research Council of Canada) and, in particular, to Dr A. Fazel Famili who became my co-supervisor. Since then, Stan and Fazel provided me with an ideal environment. They have been extremely understanding and helpful while I was initiating myself to the field of Machine Learning and, at the same time, trying to learn English. Throughout this long project, Fazel and Stan have shown unbeatable positive attitude and enthusiasm. I'm deeply thankful for their careful readings and the numerous comments they provided on earlier drafts of this thesis. I have a great debt of gratitude to both of you Stan and Fazel.

All the examiners on the committee: Dr. Rich Caruana, Dr. Robert Holte, Dr. Nathalie Japkowicz, and Dr. John Oomen provided thoughtful comments on the thesis and great suggestions for future work. I have been impressed by your dedication and I learned a lot through our discussions. A special thanks goes to Dr. Japkowicz for bringing the ICA technique to my attention a few years ago. The ICA technique played an important role in the evaluation of the method proposed in this dissertation.

I want to express my appreciation to all of my colleagues at NRC. Although, they have been forced to deal with the effects of my numerous absences during the writing phase, they always supported me and encouraged me. In particular, I want to thanks the members of the ADAM and WILDMiner projects: Chris Drummond, Fazel Famili, George Forester, Bob Orchard, Elizabeth Scarlett, Julio J. Valdés, Chunsheng Yang, and Marvin Zaluski. Bob Orchard, my group leader, generously took on my project management responsibilities

in these two projects to let me concentrate on the writing of the thesis. Bob also helped me a lot by reviewing several chapters of the thesis. I also very much appreciated the support from my previous group leader, Mike Halasz. The director of our institute, Dr. Andrew Woodsworth, never missed an opportunity to manifest his support. I'm obliged to all of you for your patience, comprehension, and encouragements.

Mon épouse, Caroline, est la personne la plus importante de ma vie. Caroline est avec-moi depuis la fin du secondaire et c'est avec elle que j'ai pris les décisions les plus importantes et relevé les défis qui les accompagnent. Le doctorat, tout comme la famille et les autres projets, c'est ensemble que nous l'avons fait. Caroline, je te remercie sincèrement pour ton amour. J'ai aussi une pensée spéciale pour nos enfants Maxime, Guillaume, et Kamille. Je suis émerveillé par la patience et par la sagesse que vous avez démontrées durant les longues heures que papa travaillait sur le doctorat. Je vous adore mes trésors! Finallement, je veux remercier tous les membres de ma famille et ceux de la famille à Caroline pour leur confiance et leurs encouragements soutenus.

This research has been supported by scholarships from NSERC (National Sciences and Engineering Research Council of Canada) and FCAR (Fonds pour la Formation des Chercheurs et Aide à la Recherche).

Chapter 1

Introduction

1.1 Problem statement

Supervised machine learning is concerned with the study of computer systems that learn models from a collection of instances. Each instance is described by a number of attributes $X_1, X_2, \ldots, X_p$ and a class attribute C. The goal is to learn a model that predicts the value of C based on the values of the other attributes. Examples of supervised machine learning techniques to learn such a model include: decision trees, neural networks, and k-nearest neighbors. Regardless of the technique used, it has long been recognized that one of the best ways to optimize performance is to improve the quality of the input attributes $X_1, X_2, \ldots, X_p$ [Michalski, 1983; Dietterich et al., 1982; Lenat & Brown, 1984].

Examples of problematic attributes which may have a significant negative impact on the performance are: irrelevant attributes, redundant attributes, randomly correlated attributes with the label, and low-level interacting attributes. Irrelevant, redundant, and randomly correlated attributes are useless and removing them from the analysis can improve the results. A variety of feature selection techniques now exists to automatically identify and remove these useless attributes [Liu & Motoda, 1998]. On the other hand, one cannot simply ignore the low-level interacting attributes during the analysis because they potentially contain useful information for classification.

As we illustrate in the next section, attribute interactions may increase the complexity of the learning task by *dispersing* the instances that belong to the same class across the attribute space. In such cases, the initial attributes, when taken individually, appear to be only remotely related to the class attribute. To uncover the predictive power of the initial attributes, the learning systems need to analyse the interacting attributes simultaneously and then fit an appropriate model for the types of interaction observed. This is a complex task that surpasses the ability of most existing machine learning systems.

Several researchers explained the disability of current learning techniques to handle attribute interactions. In particular, Rendell & Seshu [1990] emphasizes the fact that current machine-learning techniques rely on the assumption of simple attribute interactions

which make them sub-optimal in domains with important attribute interactions. Focusing on the attribute evaluation process, Kononenko & Hong [1997] and Bloedorn & Michalski [1998] explain that all learning approaches that evaluate the usefulness of each attribute individually using quality measures such as the information gain, the gini-index, the distance measure, or the j-measure are likely to generate inaccurate or too complex models whenever there are important attribute interactions. There have been numerous works on the naive-Bayes classifier (also called Simple Bayes classifier) [Duda & Hart, 1973], a well-known learning technique that assumes complete attribute independence [e.g., Kononenko, 1991; Langley, 1993; Domingos & Pazzani, 1997]. Specific issues such as the *replication* and the *fragmentation* problems with decision trees are also directly related to the lack of capacity of current techniques to deal with attribute interactions [Zheng, 1996; Vilata et al., 1997].

The problem of attribute interactions is particularly important in Knowledge Discovery in Databases (KDD) applications in which machine learning techniques are used to extract patterns from real world data [Fayyad et al., 1995]. This is further illustrated in a recent review by Freitas [2001] which explains that attribute interactions is the norm in KDD applications and failing to address this problem adequately has important consequences on the performance obtained. One approach to obtain adequate results is to reformulate the initial data representation into a more appropriate one for the learning task. As observed by Langley & Simon [1995], in most successful applications of machine learning, it is the users that have to manually perform this complex and crucial task because appropriate tools are not yet available. In addition to be difficult, manual data transformations are generally time consuming and very expensive [e.g., Quinlan, 1983; Fayyad et al., 1993].

The lack of capacity of current techniques with respect to attribute interactions combined with the practical needs of KDD applications have fostered significant research in the area of constructive induction. Constructive induction is a sub-field of machine learning that researches and develops techniques to enhance the power of the data representation used in learning. Many aspects of constructive induction have been studied over the last 15 years [e.g., Matheus, 1989; Fawcett, 1993; Bloedorn, 1996; Donoho, 1996; Zheng, 1996; Hu, 1999; Jakulin, 2002]. Some techniques have been proposed to generate new features for domains in which there are important attribute interactions. As we will discuss in the Chapter 2, the existing techniques are limited to attribute interactions between nominal or binary attributes. This dissertation focuses on the generation of continuous features that account for the interactions involving continuous attributes, which include interactions between nominal (or binary) attributes and continuous attributes as well as interactions between continuous attributes only.

1.2 Attribute interactions and relevance

The concept of attribute **relevance** plays a central role in this research. As pointed out by Mizzaro [1997], the term relevance has been used extensively in the literature and bears many meanings. In this research, we use the concept of relevance or attribute relevance to designate the usefulness of a given attribute to predict the values of the class attribute. Unless noted otherwise, we assume that relevance is computed through a univariate measure such as the information gain [Quinlan, 1983], the gain ratio [Quinlan, 1986], or the χ^2 [Kass, 1980; Mingers, 1989] measures. Moreover, we use the terms *local* and *global* attribute relevance to indicate the scope of the relevance . In particular, a locally relevant attribute is relevant in sub portions of the instance space only, while a globally relevant attribute is relevant over the full instance space.

To illustrate the links between attribute interactions and the potential benefits of globally relevant features, let us consider two simple examples. The first one involves continuous attributes only, while the second example involves interactions between a nominal attribute and a continuous attribute. The two examples represent binary classification tasks.

Example 1 Let us suppose a dataset with three attributes X_1, X_2, and X_3 that follow a multivariate normal distribution defined with the following parameters:

$$\mathbf{u_0} = \begin{bmatrix} 20 \\ 30 \\ 15 \end{bmatrix} \qquad \mathbf{u_1} = \begin{bmatrix} 20 \\ 30 \\ 25 \end{bmatrix}$$

$$\mathbf{\Sigma} = \mathbf{\Sigma_0} = \mathbf{\Sigma_1} = \begin{bmatrix} 650.0 & 0 & -225 \\ 0 & 50 & -115 \\ -225 & -115 & 350 \end{bmatrix}$$

where $\mathbf{u_0}$ and $\mathbf{u_1}$ are the class-conditioned mean values for X_1, X_2, X_3 and $\mathbf{\Sigma}$ is the variance-covariance matrix (same for both class values). We notice that attribute X_3 is the only relevant attribute for this problem since the other two follow exactly the same distribution regardless of the class value. As indicated by $\mathbf{\Sigma}$, X_3 also interacts with the attributes X_1 and X_2. Figure 1.1 shows a dataset of 100 instances generated from the above distributions. As seen from the scatter plots of X_3 versus X_1 and X_3 versus X_2 (Figures 1.1 (a) and (b), respectively), it is difficult to separate the positive from the negative instances. This difficulty is further illustrated by the class-conditional density curves shown in Figure 1.1 (c); the great overlap between the two curves clearly indicates that any decisions based on X_3

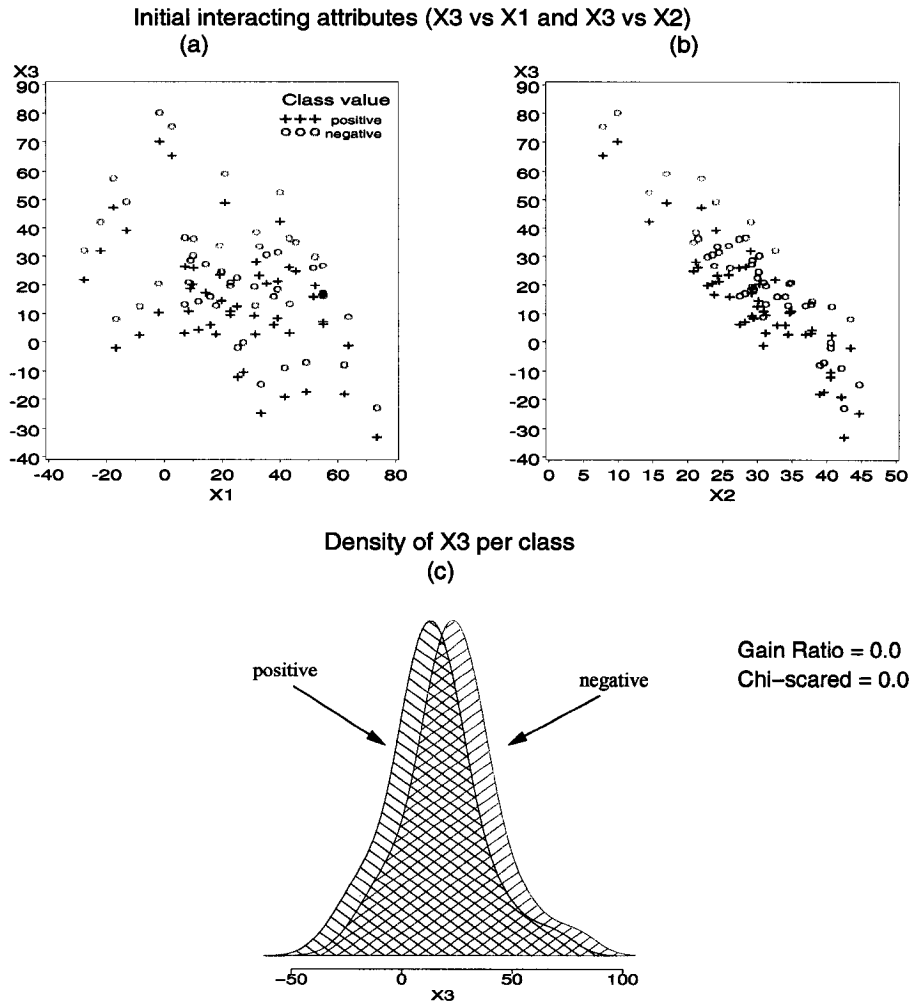


Figure 1.1: A dataset with continuous low-level interacting attributes.

will be highly error-proned. The null gain ratio and χ^2 values confirm that X_3 appears powerless in predicting the class values.

To remove the effects of X_1 and X_2 on X_3, we could generate a new feature NEW_X_3 as

$$NEW\_X_3 = 0.125 * X_1 + 0.833 * X_2 + 0.362 * X_3 \qquad \square$$

where the coefficients are taken from the third row of the inverse of the root matrix for $\mathbf{\Sigma}$[1]. The new feature contains the information from X_3 without being correlated with X_1 and

[1]We explain how to derive such transformations in Section 3.4.

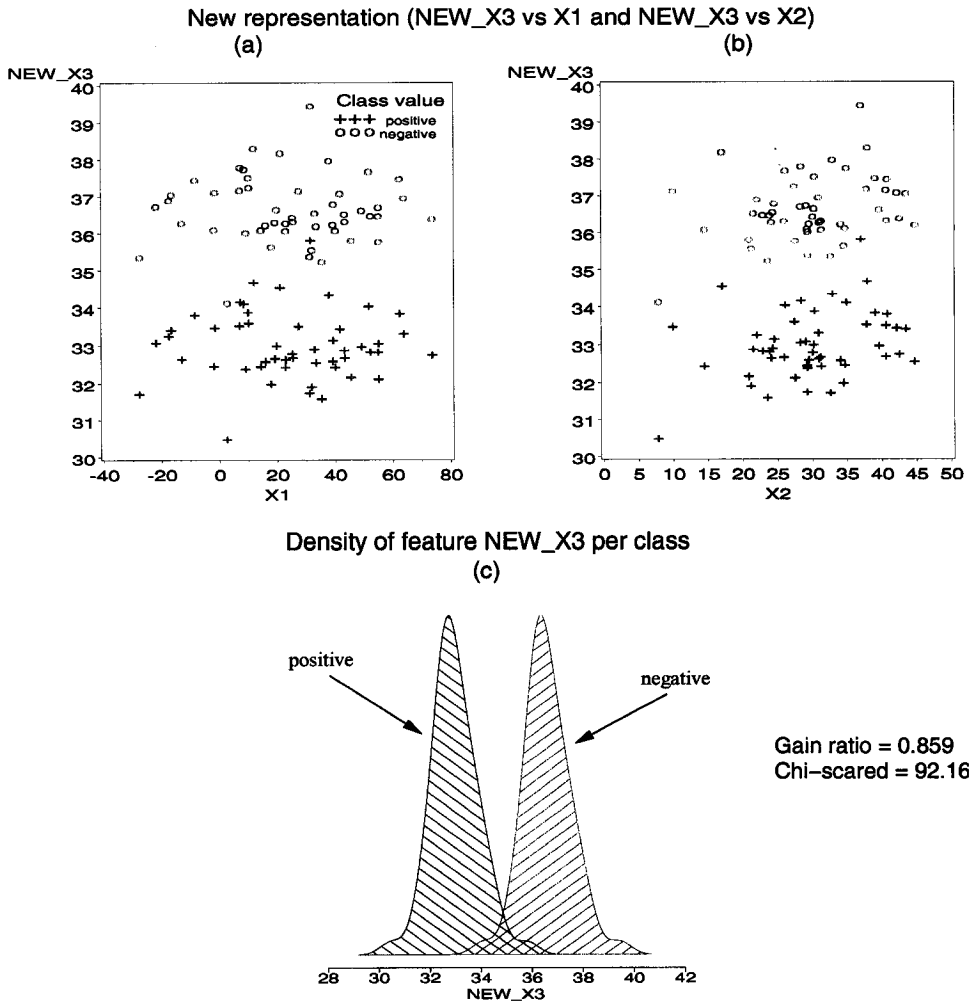


Figure 1.2: A globally relevant feature for the dataset shown in Figure 1.1.

X_2. Figure 1.2 (a) and (b) present the new feature versus X_1 and X_2, respectively. We observe that the transformation removed a great proportion of the initial dispersion since the instances of the same class are now grouped together. As illustrated by Figure 1.2 (c), the new feature NEW_X_3 is much more relevant than the initial attribute X_3. The new feature is also globally relevant since its power is observable across the full dataset. By opposition to the initial attributes, the new feature shows a great potential for classification even if it is analyzed independently of the other ones. This property facilitates the learning and increases the likelihood of success of methods that assume attribute independence.

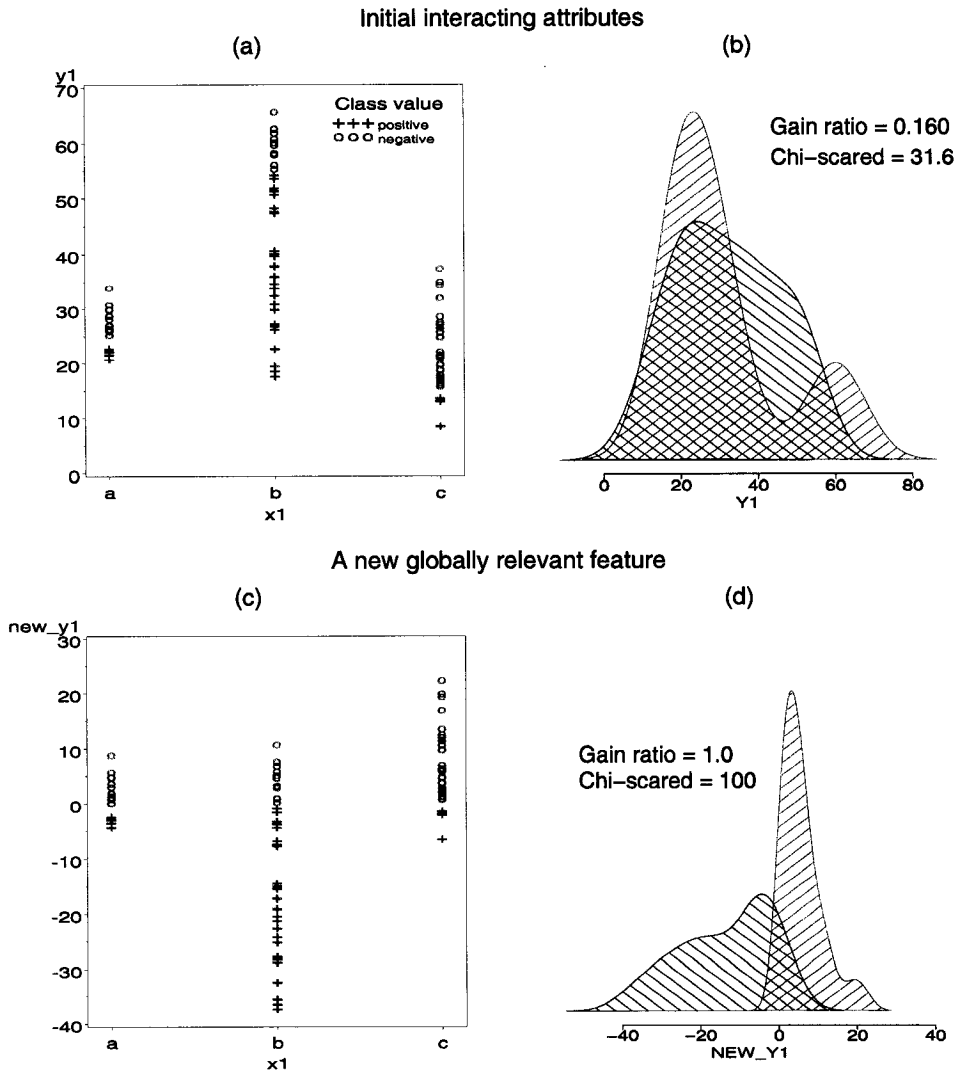


Figure 1.3: A dataset with interactions between a continuous and a nominal attributes and a globally relevant feature that accounts for the effects of the initial interactions.

Example 2 As a second example, let us consider the dataset shown in Figure 1.3 (a). We observe that the nominal attribute X_1 interacts with the continuous attribute Y_1. We also notice that, within each group, Y_1 can be used to perfectly separate the positive and the negative examples. However, when taken individually (i.e., without considering X_1), Y_1 does not appear to be relevant since the two class-conditional density curves in Figure 1.3 (b) greatly overlap and the gain ratio and χ^2 measures are relatively low. In other words, Y_1 is locally relevant but not globally relevant.

Figure 1.3 (c) and (d) displays the scatter plot and the density curves for a new feature New_Y_1 defined as

$$NEW\_Y_1 = \begin{cases} Y_1 - 25.0 & \text{if } X_1 = a \\ Y_1 - 55.0 & \text{if } X_1 = b \\ Y_1 - 15.0 & \text{if } X_1 = c \end{cases} \quad \square$$

This transformation increases the global relevance by cancelling the effects of X_1. Both mesures of relevance clearly report the high predictive power of the new feature. Accordingly, the new feature can be analyzed independently of the other attributes, which should simplify the learning task and improve the accuracy of the model learned.

In both examples, the initial relationships between the attributes decrease the global relevance. However, it is also possible that some interactions have a positive effect on the global relevance. In such cases, it is best not to transform the initial data. Before attempting a transformation to cancel the effects of a given interaction, it is important to assess the actual effect of this interaction on the learning problem at hand. To resolve this issue, this dissertation introduces a technique to evaluate the interactions observed in terms of their effects on the global relevance. Using this information, the transformation process can focus on the interactions that negatively affect the learning process. The data transformation approach we propose in this dissertation produces transformations like the ones presented in the examples above as well as more complex transformations involving many attributes. The technique is also general in the sense that it requires no information about the underlying distributions of the data. The framework proposed allows the user to participate in both the identification of interactions and the construction of new features.

1.3 Related areas of research

Figure 1.4 provides an overview of research areas related to learning with attribute interactions. There are two types of solutions. The first one, as advocated above, is to develop a new representation that is appropriate for the existing learning systems. The second type of solution focuses on the learning methods. There are already learning methods that try to account for interactions by analyzing several attributes simultaneously. Examples include the SVM and NN methods. These methods have other drawbacks and are not guaranteed to properly handle all kinds of interactions but they are known to be generally more robust with respect to attribute interactions. One might also try to enhance existing methods by adding the required functionality to deal with attribute interactions. Such enhancements

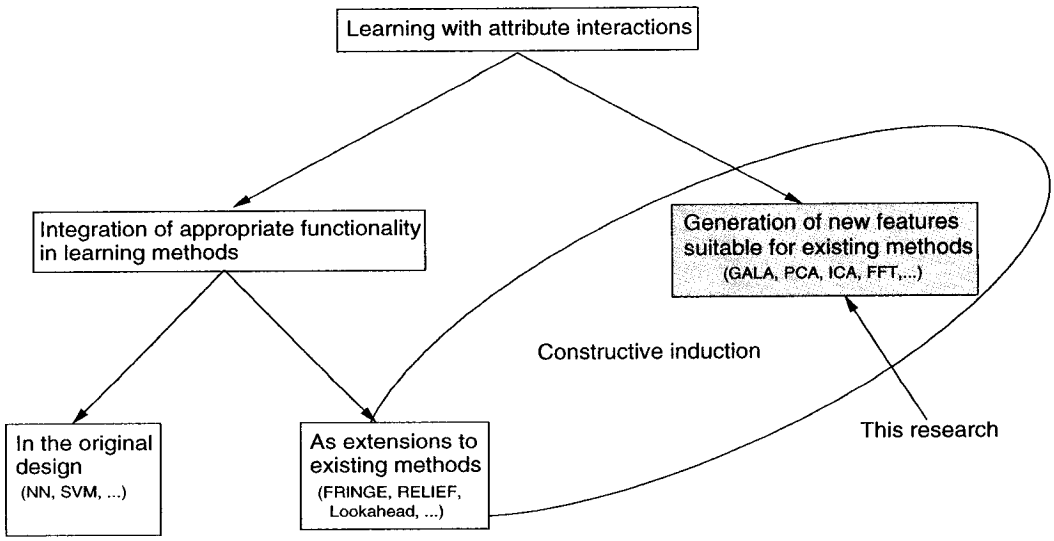


Figure 1.4: Related research.

may or may not include internal data re-representation mechanisms.

The field of constructive induction includes all research related to the data representation. This is illustrated in Figure 1.4 with the oval that encloses all research on generation of new features and part of the research on enhancement of existing learning systems (the enhancements accomplished through data re-representation). These two types of research follow different objectives. The research on generation of new features targets the creation of a new dataset that could be used independently at a later stage for learning or for other purposes. On the other hand, the research on enhancements of existing methods may create new features during the learning process but without the intent of using these new features independently. Accordingly, they do not generate a new dataset.

As indicated in Figure 1.4, this research focuses on the creation of a new representation, in particular, a representation that is helpful with respect to the current learning task. As we shall discuss in the literature review (Chapter 2), existing constructive induction techniques are limited to boolean attributes or nominal attributes and very little research has been performed on the generation of continuous attributes. Also discussed in Chapter 2, the existing statistical techniques to deal with continuous attributes mostly ignore the class attribute which make them sub-optimal with respect to classification learning tasks. This research addresses an important need by providing techniques to analyse interactions involving continuous attributes in terms of their effects on learning and by generating new globally relevant continuous features that can be used with any standard learning system.

We use the name GLOREF (GLObally RElevant Features) to designate the approach that integrates the proposed techniques.

1.4 Thesis statement and Contributions

The central idea of this dissertation is:

> *Augmenting the data representation with globally relevant continuous features, derived from the analysis of interactions between nominal and continuous attributes (or only between continuous attributes), improves the performance of several learning systems.*

and the contributions are:

- A method and a large-scale experiment to study the effects of attribute interactions in learning.
- A novel method to assess attribute interactions that takes into account the effects on learning.
- An interactive tool to visualize the effects of interactions and to explore possible transformations.
- A new data transformation technique that generates globally relevant continuous features.

The conclusion chapter describes these contributions in more details.

1.5 Structure of the dissertation

This dissertation has six additional chapters. The next chapter discusses related work and identifies the need for this research.

Chapter 3 proposes an empirical evaluation of the robustness of various learning algorithms with respect to attribute interactions. The study presented also provides an estimation of the potential benefits of globally relevant features. The chapter concludes by presenting the challenges of this research.

Chapters 4 and 5 describe the GLOREF approach. First, Chapter 4 introduces the method to analyse the attribute interactions in terms of their effects on learning. Chapter 4

also presents the *Relevance Graphs* and *Interactive Relevance Graphs* which are two techniques we have developed to visualize the effects of the interactions and investigate potential transformations. Chapter 5 describes how the information computed in Chapter 4 is used to generate the new features. The feature generation method produces globally relevant continuous features that account for univariate and multivariate relationships.

Chapter 6 proposes an extensive evaluation of the GLOREF approach. The experiment includes the artificial datasets described in Chapter 3 and 12 real world datasets publicly available. This chapter also compares the GLOREF approach with two other approaches (Principal Component Analysis and Independent Component Analysis).

Chapter 7 summarizes the dissertation and discusses future work.

A list of notations and a glossary are provided at the end of the dissertation.

Chapter 2

Literature Review

2.1 Introduction

As explained in Section 1.3, previous attempts to learn from domains with important attribute interactions can be broadly classified in two categories: those that incorporate appropriate functionality in the existing learning methods and those that transform the data representation. In turn, extensions to existing methods may or may not involve constructive induction. We first review enhancements proposed to enhance existing methods that do not rely on constructive induction. Then we discuss research on enhancing the data representation.

2.2 Enhancements of existing learning approaches

This section reviews enhancements proposed to improve the robustness of existing learning techniques without modifying the initial data representation. The enhancements discussed relate to the decision tree and the naive-Bayes learning approaches.

2.2.1 Improving decision trees

One of the most important tasks in the decision tree learning algorithm is the selection of the attributes used to partition the data in the nodes of the tree. Commonly-used heuristics for this task assume that the attributes are conditionally independent given the class value and evaluate the usefulness of each attribute independently. This approach is efficient but may lead to complex and inaccurate trees in domains with strong attribute interactions. One possible strategy to solve this problem is to use a more sophisticated measure that evaluates each attribute in the context of the other attributes. The next two sub-sections describe techniques that implement this strategy.

Lookahead mechanism

The idea of the lookahead mechanism is to select for a given node the attribute that will lead to the most suitable tree at some further point in the process. This strategy allows the selection of an attribute that does not optimally partition the current set given the promise that it will eventually lead to a better tree. Interacting attributes that need to be considered together in order to be useful for classification are therefore more likely to be selected. Unfortunately, the complexity of the search increases exponentially with the number of lookahead steps. In most applications, it is not practical to look at more than two levels ahead. This prevents the detection of relationships involving several attributes.

Murthy & Salzberg [1995] analyzed empirically the usefulness of the limited lookahead strategy. They concluded that limited lookahead is not particularly helpful. Furthermore, they reported that limited lookahead can lead to inferior results in a number of applications.

Local measures

Commonly used impurity measures such as the information-gain-ratio [Quinlan, 1993] and gini-index [Breiman et al., 1984] are named *global* measures since they estimate the quality of a given attribute using all instances in a given partition. An alternative approach is to perform this evaluation based on subgroups of instances. This alternative approach works as follows. First, it divides the instance space into sub-regions and then it estimates the quality of a given attribute by measuring its ability to discriminate instances that belong to the same sub-region. The idea is that a good attribute for classification should be able to separate the instances that are similar (i.e., that are close to each other in the instance space). Since the measure of usefulness is limited to part of the data, these measures are named *local measures*. Local measures are expected to be more robust against the problem of attribute interactions because each attribute evaluation takes into account several attributes. The RELIEF family of systems [Kira & Rendell, 1992; Kononenko, 1994; Robnik-Šikonja & Kononenko, 1997] and the CI measure [Hong, 1997] propose two different local attribute quality measures. These approaches can be used to perform feature selection in a pre-processing step or inside a decision tree system as a replacement of the standard attribute evaluation mechanism. The last strategy seems to be more promising, in particular when the target is a symmetric function of some attributes (e.g., a majority function) [Kononenko & Šimec, 1995].

We note that the strategy we follow in this research goes into the opposite direction. Instead of trying to adapt the measure to the *local relevance* of the data, we try to make

the data *globally relevant* through appropriate transformations such that global measures of relevance lead to accurate results.

2.2.2 Improving naive-Bayes

Naive-Bayes is a natural candidate to consider when studying the problem of attribute interactions. The assumption of independence plays a key role in the algorithm and it is commonly thought that naive-Bayes is extremely sensitive to attribute interactions. Accordingly, a number of researchers proposed extensions to naive-Bayes in order to reduce the effects of attribute interactions.

A first approach is to use the technique of feature subset selection. Langley & Sage [1994] experimented with a forward feature selection technique and six datasets from the UCI repository [C. Blake & Merz, 1998]. They observed some improvements on three of them. However, it is not clear that there were significant attribute interactions in any of the datasets used. Pazzani [1995] also used feature selection but in conjunction with feature joining. In Kononenko [1991], the attributes are first partitioned into groups and independence is assumed only between attributes from different groups. Experimental results provided by Kononenko [1991] showed slight improvements only.

Other approaches focus on the extension of the algorithm itself. Langley [1993] suggests dividing the instance space in sub-regions and building a naive-Bayes classifier for each of these regions. Kohavi [1996] introduces the NBTree algorithm which learns a decision tree with a naive-Bayes classifier for each leaf node. The author argues that this combination is particularly useful with large data sets. Less restrictive forms of naive-Bayes have also been explored [Sahami, 1996; Cheng & Greiner, 1999; Friedman et al., 1997].

The experiment we propose in Chapter 3 allows us to evaluate the sensitivity of the naive-Bayes with respect to attribute interactions and to compare it with other learning approaches. The results confirm that the naive-Bayes is indeed sensitive to attribute interactions but not necessarily more sensitive than other popular machine learning techniques.

2.3 Constructive induction

Michalski [1983] introduced the term *constructive induction* (CI) to describe the process by which new potentially more powerful features are added to the representation space. Other names for constructive induction include: feature generation, feature construction, feature extraction, and bias shift [Utgoff, 1986]. Recent constructive induction frameworks extend

the initial definition by including both the addition and the removal of attributes [Fawcett, 1993; Wnek & Michalski, 1994; Bloedorn & Michalski, 1996].

Constructive induction systems may be designed to solve different problems with the representation space. Two examples are the problem of indirectly relevant attributes [Bloedorn & Michalski, 1998; Rendell & Ragavan, 1993] and the problem of low level representation [Rendell & Seshu, 1990]. Constructive induction systems address these problems by trying to automatically create new features that combine the initial attributes. The aim is to find features that are more appropriate for learning than the initial attributes. As explained in Section 1.3, constructive induction might be integrated with a given learning approach to extend its initial ability or developed as a separate technique to transform the data representation independently of the learning system used to construct the models. Although this research fits in the later category, we also review related work on enhancement of existing learning approaches.

Our discussion is divided into three sections. The first one discusses constructive induction techniques developed to deal with the problem of contextual normalization. The second section proposes a classification of constructive induction techniques based on their use of information about attribute interactions. The last section summarizes the various techniques discussed based on their characteristics.

2.3.1 Contextual normalization

The problem of attribute interactions is related to the notion of context which received significant attention over the last decade in the machine learning community [Widmer & Kubat, 1998; Matwin & Kubat, 1996]. Context can be informally defined as any information that surrounds the event or object of interest. The context is usually not involved in the final model but may help improving performance by complementing the information that is directly relevant.

Most of the research on learning with context deals with applications for which the contextual information is known to be important, but not made explicit to the learner. These applications require systems that can detect hidden changes and adapt the classification strategy accordingly. Examples are the FLORA family of systems [Widmer & Kubat, 1996] and the SPLICE system [Harries et al., 1998]. Our focus is on applications in which the context is partially available through at least one of the initial attributes. This situation is not uncommon; examples are from speech recognition, medical, and engineering applications in which we have attributes such as sex, age, living area, ambient temperature, pressure, and humidity. These attributes are often named *contextual attributes* by opposition to *primary*

or *relevant attributes* which contain more task specific information. Contextual normalization [Katz et al., 1990; Turney & Halasz, 1993] tries to improve the representation by cancelling the effects of the contextual attributes.

Contextual normalization plays a key role in several real world applications. For instance, in medical domains, normalization techniques are used, among others, with electromyographic data [Knutson et al., 1994], VO_2 data [Toth et al., 1993], and magnetic resonance data [Sled et al., 1998]. Nonlinear normalization is utilized in character recognition [Lee & Park, 1994] and speaker normalization [Watrous, 1993]. Normalization is also applied to reduce variability in pollution data [Delbeke et al., 1995] as well as in financial [Horrigan, 1978] and industrial applications [Sola & Sevilla, 1997]. Although the normalization techniques vary considerably from one application to another, the objective is common; transforming the initial measurements in order to take into account the interactions with the contextual attributes.

Researchers in the machine learning community have proposed frameworks and techniques for domain independent contextual normalization. The next sub-section describes these contributions. The following two sub-sections discuss the related issue of the automatic identification of contextual attributes and some alternative approaches to use contextual information.

Eliminating the effects of contextual attributes

The goal of contextual normalization is to transform the relevant attributes such that the influence of the contextual attributes is canceled. The advantage is that the new normalized relevant attributes allow meaningful comparisons between the observations or subjects. Only the new normalized attributes should be needed for learning. The models obtained from the new attributes should be more general than the ones obtained without normalization, since they are independent from the variations in the contextual attributes.

Donoho and Rendell Donoho & Rendell [1996] explain that contextual normalization is generally very difficult to perform because it requires in-depth knowledge of the interactions between the contextual attributes and the relevant attributes. They propose the use of ratios to address the problem of attribute normalization. In their approach, each relevant attribute is divided by the product of all potentially influencing contextual attributes. The ratios obtained are used instead of the initial attributes during learning. Background knowledge is used to determine which contextual attributes influence a given relevant attribute. The authors suggest using this approach only in domains where more sophisticated normalization

formulas cannot be obtained from domain experts [see Donoho & Rendell, 1996, page 119].

Turney and Halasz Turney & Halasz [1993] describe a contextual normalization technique and apply it to the diagnosis of aircraft engine faults. The normalization proposed transform each relevant attribute x_i into a new attribute x'_i using the following formula

$$x'_i = \frac{x_i - \mu_i(C)}{\sigma_i(C)}$$

where $\mu_i(C)$ and $\sigma_i(C)$ represent the expected value and expected variation of x_i given the values of contextual attributes in C, respectively. The authors propose two techniques to estimate the parameters $\mu_i(C)$ and $\sigma_i(C)$. One relies on instance-based learning and the other is based on multiple linear regression. Both techniques require the availability of baseline instances (i.e., instances representing healthy engines) collected over a wide range of ambient conditions (i.e., context).

With the instance-based approach, $\mu_i(C)$ is equal to the mean of x_i considering only the k closest baseline instances. The closest instances are found using the contextual attributes in C and a standardized distance measure. To estimate $\sigma_i(C)$, the algorithm finds the next k_2 closest instances and, for each of these instances, it computes the root squared difference between x_i and $\mu_i(C)$. Then $\sigma_i(C)$ is set to the average of these differences.

With the multivariate linear regression approach, a global linear model is built for each relevant attribute x_i. The $\mu_i(C)$ are the response variables and the contextual attributes in C are the predictors. Step-wise variable selection is used to discard unrelevant contextual attributes. They use a standard technique in linear regression for calculating variation [Fraser, 1978]. The authors report that the instance-based and multivariate linear regression normalization approaches lead to similar results.

Letourneau and al. In Létourneau et al. [1998], we have observed that the existing contextual normalization techniques are very restricted. In particular, the use of ratios as proposed by Donoho & Rendell [1996] is not flexible enough to take into account the various forms of interactions that may exist. The two approaches by Turney & Halasz [1993] are more powerful but are restricted to domains in which one can find appropriate baseline instances covering a wide range of contextual conditions. Moreover, none of the techniques can be applied if there is no prior knowledge to identify the contextual attributes. Accordingly, we have proposed a new technique that automatically identifies the contextual attributes and derives transformations that remove the effects of the contextual attributes

on the relevant attributes. The proposed approach works as a pre-processor and can be used with any standard learning algorithms. The approach relies on the ANOVA to assess the interactions along with particular discretization and clustering techniques. The approach showed great potential in reducing the variability in commercial aircraft engine data. However, when applied to the task of predicting engine component failures [Létourneau et al., 1999], the new features did not seem to help improve the accuracy. The fact that the approach does not use the class information may explain the obtained result. The techniques presented in this dissertation overcome the above limitations by focusing on the generation of highly relevant features.

Identification of contextual attributes

One of the prerequisites to the elaboration of normalization models is the identification of the contextual and relevant attributes. This is typically done using domain knowledge. Automatic methods are highly desirable for this task because they may help simplifying the elaboration of normalization formulas. They may also lead to the identification of new contextual attributes. Turney [1996a] approaches this problem by providing formal definitions for primary and contextual attributes. These definitions are based on earlier work by John et al. [1994] on the problem of feature subset selection. The author proposes an adaptation of the formal definitions in order to make them applicable in practice. However, the actual usefulness of the approach has not been demonstrated using real world applications. We note that the contextual normalization approach introduced by Létourneau et al. [1998] can automatically determine the contextual attributes but can also use knowledge of attribute's role (i.e., contextual or primary) to reduce computations.

Alternative approaches to use contextual attributes

There are four approaches that can be used in addition to contextual normalization to exploit contextual attributes [Katz et al., 1990; Turney, 1996b]:

Contextual expansion The contextual attributes are used to expand the representation space. The learning algorithm treats them as any other attributes.

Contextual classifier selection Classifiers are build for each context. During classification, a meta level mechanism is used to decide which classifier(s) should be used based on the values of the contextual attributes. Various voting schemes can be used to efficiently combine results from more than one classifier.

Katz et al. [1990] uses classifier selection with neural networks and shows improvements on an automatic target recognition problem. This technique has also been used with decision trees in speech recognition [Kubat, 1996] and tonal music harmonization [Widmer, 1996].

Contextual classifier adjustment The contextual attributes can be used in a post-processing step to adjust the prediction made by the classifier. [Watrous, 1991] describes such an approach with decision trees.

Contextual weighting The relevance of each attribute may be a function of context. Contextual weighting is used by Turney [1993] in a vowel recognition task.

Combinations of these approaches may also lead to interesting results. Turney [1993] obtained significant improvements in a vowel recognition task by combining contextual normalization, contextual expansion, and contextual weighting. Kubat [1996] successfully combines contextual selection and contextual adjustment.

2.3.2 Use of attribute interactions in Constructive Induction

Constructive induction systems are generally not designed for domains in which there are significant interactions among the attributes [Pérez & Rendell, 1996b]. However, information about attribute interactions may be used to guide the search towards meaningful attribute combinations. Because attribute interactions are the central theme of this research, we emphasize the role of attribute interactions in various constructive induction approaches. The next four sections examine constructive induction from the following perspectives: techniques that do not consider attribute interactions, techniques that consider attribute interactions at the knowledge level, techniques that consider attribute interactions at the model level, and finally, techniques that exploit attribute interactions found at the data level.

Without considering attribute interactions

The methods described in this section are the ones that do not rely on any information about attribute interactions during the search for new useful features. These approaches typically need some forms of heuristics to constrain the search space. For example, they may use only a very small number of operators and/or consider only few attributes for potential combinations.

AQ17-DCI The AQ17-DCI [Bloedorn & Michalski, 1998] system combines a data-driven constructive induction method and a rule learning system. The data-driven constructive induction method can create new attributes and remove irrelevant ones. It includes relational, arithmetic, and various functional operators (e.g., minimum, average, count). The user can select a subset of operators to apply among an initial list provided by the system. AQ17-DCI exhaustively applies the selected operator(s) to combine the attributes. Only the new attributes that have an information gain ratio [Quinlan, 1993] greater than a user specified thresholds are kept for learning. AQ17-DCI performs discretization of numerical attributes using the ChiMerge approach [Kerber, 1992]. The output of AQ17-DCI is a set of rules and not a new representation to be used independently. The type of features it produces are limited by the set of operators implemented and augmenting the set of operators is likely to make the search intractable.

GALA GALA [Hu, 1999] is a pre-processing system that can be used with any standard learning algorithm. The first version of GALA creates boolean features. GALA follows an iterative approach. At each step it generates new boolean attributes by combining the existing ones. These attributes are then evaluated using a relative measure. Only the attributes that lead to a significant increase in gain ratio with respect to their parents will be kept for the next step. The author reports that the attributes generated by GALA can improve the performance of C4.5 [Quinlan, 1993], CN2 [Clark & Boswell, 1991] and various neural networks. Some of GALA's limitations are: (i) it uses only two operators "AND" and "NOT", (ii) it needs to "booleanize" the input attributes, and (iii) cannot remove useless attributes. On the other hand, the simplicity of the operations ensures that the new features are comprehensible. The second version of the system, GALA II, is designed to learn more complex concepts such as the m-of-n and majority rules as well as the binary features learned by the first version of GALA. GALA II uses the multi-layer neural network technique to extract the prototypical structures. Once the neural network is trained, the features are extracted and presented in a comprehensible manner. Although it uses the neural network technique, GALA II is not designed to recognize various forms of relationships between continuous attributes.

STAGGER STAGGER [Schlimmer, 1987] uses constructive induction in applications for which its concept description language does not allow perfect predictions. STAGGER uses a set of feature/value pairs to describe the target concept. A weight is assigned to each element in the set. During classification, these weights are combined to compute the

predicted class. The weights are updated after each classification in order to reflect the value of each feature/value pair. Whenever there is a prediction error, STAGGER creates new attributes by combining the existing ones with boolean operators (i.e., "NOT", "AND", and "OR"). The new feature will then compete with the initial ones. In the long run, only the new attributes that outperform the initial ones will stay in the description of the concept. We note that the output of STAGGER is a description language and not a new data representation that can be used for learning.

Attribute creation in decision trees There has been a number of efforts on trying to incorporate the creation of linear combinations of attributes with the decision tree learning approach. OC1 [Murthy et al., 1994], CART [Breiman et al., 1984], and LMDT [Utgoff & Brodley, 1990] are examples of systems that perform this integration. At each node, these systems search and evaluate various linear combinations of attributes. If one of these combinations appears to be more powerful than the initial attributes, then it can be used as test attribute for that node. Limiting the search to linear combinations allows a reasonable efficiency. None of these systems explicitly rely on information about attribute interactions to guide the search. Like AQ17-DCI, all of these systems extend existing algorithms and are not generating new features to use with other learning systems.

Considering attribute interactions at the knowledge level

Constructive induction methods can exploit various forms of domain information provided by the user to guide the search. The format and extent of domain knowledge vary considerably from one application to another. In some applications, complete domain theories exist and can be used to construct new relevant attributes that have high classification power. Examples of systems that can use complete domain theories include LAIR [Elio & Watanabe, 1991], TGCI [Donoho & Rendell, 1995], and AQ15 [Michalski et al., 1986].

In practice, complete theories are rarely available. What is available are fragments of knowledge. Donoho [1996] examines twelve types of knowledge that might be available in real world applications and discuss how they can help in the creation of new features. Three of them are related to the problem of attribute interactions:

Contingency knowledge Contingency knowledge gives information that the effect of one feature is contingent upon the value of another feature. This highlights the relationships between the attributes and may help, for instance, identifying the contextual and primary attributes.

Normalization knowledge Normalization knowledge provides information about the effect of a group of intermediate concepts (some relevant, others not) on a given relevant attribute. This knowledge can be used to normalize the given attribute to eliminate correlations with irrelevant intermediate concepts. The techniques described in Section 2.3.1 are examples of constructive induction methods that rely on normalization knowledge. We also note that contingency knowledge provides intermediate information useful for the development of normalization knowledge.

Dimensional analysis Dimensional analysis may prevent creation of illegal feature combinations. For instance, one would not try to add an attribute with unit *meters* to another attribute with unit *kilograms*. Knowledge about units may also help identifying attributes that may depend on other attributes. For instance, it is quite reasonable to suspect attributes with unit *Celsius* (a temperature) to be related to other dimensions such as geographical location, and time of the day. The COPER [Kokar, 1986] system, a system for the discovery of empirical laws, uses dimensional information to guide constructive induction.

Considering attribute interactions at the model level

They are systems that try to identify potentially useful attribute interactions by analyzing patterns in previously induced hypotheses. These systems are tied to a specific hypothesis language. We discuss four systems, the first two are for decision trees while the last two work with decision rules.

FRINGE and CITRE FRINGE [Pagallo & Haussler, 1990] and CITRE [Matheus & Rendell, 1989; Matheus, 1989] are two systems that create new attributes based on the analysis of previously induced decision trees. An initial tree is first induced using the initial attributes. The focus is on the paths that lead to positive leaves. The idea is to create new attributes that combine the attributes that appear on each of these paths. The new attributes are added to the existing attributes. There are several versions of FRINGE (i.e., Symmetric FRINGE [Pagallo, 1990] and DCFringe [Yang et al., 1991]), each of them uses a specific heuristic to combine the attributes on a given path. The simplest strategy creates a new attribute that is equal to the conjunction of the last two tests on a given path. FRINGE appears to be useful only in a small number of domains. CITRE relies on user input to decide which attributes on a path should be combined. At the limit, CITRE will create a new attribute for each pair of tests on all paths leading to positive leaves. CITRE

and FRINGE performances are similar.

CIPF CIPF [Pfahringer, 1994a] works with propositional decision rules. It is also an iterative approach that creates and adds new attributes in each step. CIPF takes a rather eager approach by generating, at each step, all possible combinations of attributes from the selected rules and a pre-defined list of operators. All new attributes are made available to the next iteration. The growing of the attribute space is controlled be a filtering mechanism applied at the end of each iteration. This mechanism analyzes the rule set and removes attributes that are not involved. CIPF keeps track of removed attributes and never introduces the same attribute again. The process stops when no new attributes can be created. The Minimum Description Length (MDL) principle [Quinlan & Rivest, 1989] is used to select the best subset of rules used to build the new attributes. The use of MDL allows CIPF to use all the data during the constructive induction process. Other measures such as accuracy would have required the splitting of the data into a test and training sets [Bloedorn et al., 1993]. CIPF combines the attributes that appear in the selected rules using various commonly used operators similar to the ones used by AQ17-DCI. CIPF 2.0 [Pfahringer, 1994b] uses C4.5 as rule inducer. CIPF 2.0 lead to some improvements over C4.5 on artificial domains with various levels of noise.

CN2-MCI CN2-MCI [Kramer, 1994] represents an alternative to CIPF. It is also designed for propositional decision rules, but works in a fairly different manner. As indicated by its name, CN2-MCI uses CN2 [Clark & Boswell, 1991] as rule inducer. Only one new attribute is created at each iteration. CN2-MCI also uses only one operator to combine the attributes. Each iteration has four steps. First, CN2-MCI analyzes the relevant rules of the existing hypothesis. Second, it selects m pairs of features that tend to co-occur in these rules (m being a user specified parameter). Third, it creates a new feature from each of these pairs. Finally, it keeps only the best of the m potential new features and adds it to the existing attributes. As in CIPF, the assessment of each hypothesis takes into account the complexity of the new features used. CN2-MCI stops after a user specified number of iterations without improvement. CN2-MCI returns the attributes representation that lead to the best hypothesis among all iterations.

Considering attribute interactions at the data level

We now examine data re-representation techniques in which the information about potential attribute interactions is found from the data itself. These techniques are particularly rele-

vant since our research assumes that such interactions exist between the initial attributes. The techniques discussed in this section have been mostly developed outside the area of constructive induction but can be used for that purpose.

BACON and FAHRENHEIT BACON [Langley et al., 1986] and FAHRENHEIT [Zembowicz & Zytkow, 1991] are two systems for automatic discovery of empirical laws. They do not address the classification task problem, but they closely rely on attribute interactions to discover useful relationships. The goal of these systems is to analyze the interactions among the attributes in order to discover empirical rules that explain the behavior of the data. There are two types of attributes: independent and dependent attributes. Independent attributes can be controlled or adjusted while dependent attributes are only observed. Both systems assume that all dependent attributes are numeric. To find relationships among a set of attributes, the systems try to simulate the scientific discovery process by performing experiments. Each experiment is designed to study the effects of a specific independent parameter in the context of the partial theory already developed. In each experiment, they systematically vary the parameter under study while holding constant the other independent terms. At the end of each experiment, the systems try to build a model explaining the behavior of the given parameter. If the obtained model is acceptable, then it is integrated in the theory under development. The process iterates until all independent parameters have been analyzed.

FAHRENHEIT extends BACON in various manners. In BACON, the data should be globally described by only one law while FAHRENHEIT allows the discovery of several equations to explain various regions of the data. The equation finder (EF) used in FAHRENHEIT allows the discovery of more complex equations than BACON. EF also adopts a more flexible treatment of errors in parameters which makes it more general.

Principal component analysis Principal component analysis (PCA) [Johnson & Wichern, 1992] is a multivariate statistical technique used in various situations (e.g., exploratory analysis, data summarization, multivariate outliers detection, and factor analysis). PCA, also known as the Karhunen-Lòeve expansion [Watanabe, 1969; Devijver & Kittler, 1982], is used as a pre-processing technique to transform the initial attributes into a new potentially smaller set of attributes. Given a dataset with p numeric attributes, PCA computes p new attributes called principal components. Each principal component is a linear combination of the original attributes with coefficients equal to the eigenvectors of the correlation or covariance matrix. Since the eigenvectors are orthogonal, the principal components represent

perpendicular directions in the original representation space. The certainty of uncorrelation between the principal components lead to believe that learning should be easier with the principal components than with the initial attributes. Moreover, the number of attributes may be reduced by selecting only the principal components that have the highest eigenvalues. This is equivalent to selecting the dimensions along which the largest variabilities are observed.

Although the principal components have many desirable properties, several researchers conclude that PCA is not particularly useful in the context of supervised classification learning [Rakotomalala, 1997; Delisle et al., 1998; Richeldi & Lanzi, 1996]. The inappropriateness of PCA may be explained by the fact that the PCA transformation is not optimal with respect to class separability [Fukunaga, 1990].

Canonical Discriminant Analysis Canonical discriminant analysis (CDA) is a data summarization technique related to PCA and canonical correlation. Like PCA, canonical discriminant analysis derives attributes that are linear combinations of the initial attributes. The attributes created using canonical discriminant analysis maximize the distance between the centroids of different classes. The number of new attributes created is equal to the minimum between the number of classes minus one and the number of attributes. Yip & Webb [1994] show how canonical discriminant analysis can be used as a constructive induction technique in classification learning. They also introduce a constructive induction technique that combines clustering and canonical discriminant analysis. This latter approach allows a greater robustness when the initial attributes are not normally distributed.

Independent Component Analysis The Independent Component Analysis (ICA) is a recent but very popular technique to discover structure in the data [Hyvärinen et al., 2001]. Like PCA, and CDA, ICA produces new features that are linear combinations of the initial ones. However, unlike the other techniques presented above, the ICA is also appropriate with non-gaussian distributions since it relies on higher-order statistics (not just the second-order statistics). The objective is to generate *independent* components (or as independent as possible). This can be seen as a generalization of the PCA's task which is to produce *uncorrelated* components. The blind source separation problem is often used to present the ICA method. Let us assume that there are three persons in a room talking independently of each others and three devices located across the room to record the *sounds* generated by the speakers. We also assume that, at any point in time, the devices record some linear combinations of the signals produced by the speakers. If we use $s_1(t)$, $s_2(t)$, and $s_3(t)$ to

denote the signals emitted by the three speakers and $x_1(t)$, $x_2(t)$, and $x_3(t)$ to denote the sounds recorded at time t, then we have

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} = \mathbf{A} \begin{bmatrix} s_1(t) \\ s_2(t) \\ s_3(t) \end{bmatrix}$$

where $\mathbf{A}$ is some unknown matrix. The ICA problem is to estimate the $s_i(t)$ and the matrix $\mathbf{A}$ given only the $x_i(t)$. This problem is resolved through an optimization routine that selects the coefficients in the matrix $\mathbf{A}$ in a way that maximizes the independence of the resulting $s_i(t)$. This explains why the $s_i(t)$ are named the *independent components*. In general, the number of components must be equal to the number of attributes observed. Also, there are situations for which there is no solution to the ICA problem. In particular, this is the case when the signals $s_i(t)$ are normally distributed. In turn, the absence of solution may prevent the optimization process from converging. In practice, one would deal with such situations by forcing the optimization procedure to stop after a given number of iterations. Finally, we note that like the PCA and CDA methods, the ICA does not rely of the class information.

2.3.3 Summary of Constructive Induction

Table 2.1 summarizes the various constructive induction systems discussed. The three systems that deal with the discovery of empirical laws (i.e., COPER, BACON, FAHRENHEIT) are not presented in this summary table since they cannot be compared with the other approaches. The first three columns display the types of information used and the last two columns characterize the types of information returned by the various approaches. Regarding the input information, the table presents the type of information about attribute interactions that is used, the type(s) of attributes that can be processed, and whether or not the class information is used. As for the output information, the table specifies if a model of the interactions between attributes is returned and the type(s) of the new attributes created (if any). To specify the type of the attributes, we use B for binary, N for nomial, and C for continuous.

Method	Input information			Output information	
	Attribute interact.	Class attribute	Attribute type(s)	Models of interact.	Feature type(s)
AQ17-DCI	None	Yes	BNC	No	None
GALA II	None	Yes	BNC	No	B
STAGGER	None	Yes	B	No	None
OC1	Model	Yes	C	No	None
CART	Model	Yes	C	No	None
LMDT	Model	Yes	C	No	None
FRINGE	Model	Yes	B	No	None
CITRE	Model	Yes	B	No	None
CIPF	Model	Yes	BNC	No	None
CN2-MCI	Model	Yes	B	No	None
Turney & Halasz [1993]	Know	No	C	No	C
Létourneau et al. [1998]	K + D	No	BNC	Yes	C
PCA	Data	No	C	Yes	C
CDA	Data	No	C	Yes	C
ICA	Data	No	C	Yes	C

Table 2.1: Summary of techniques to re-represent the data.

We first observe that a large proportion of the techniques developed are designed to be integrated with existing learning approaches and are not producing a new representation. With these systems, the focus is on the improvement of the accuracy of existing methods by opposition to be on the assessment and removal of attribute interactions. As a result, these systems do not produce a model describing the attribute interactions observed and their effects on learning. Hu [1999] noticed the lack of general data pre-processing systems that are independent of existing learning systems. Their solution was to propose the GALA systems. The GALA systems generate highly comprehensible features but the types of interactions that it can handle are limited to either prototypical relationships of boolean expressions. The GALA systems do not directly assess the interactions observed in the data and do not produce a model that describes the effects of these interactions.

The methods that can cope with continuous attributes are either from research in contextual normalization or from the statistical and pattern recognition fields. These methods tackle the problem of attribute interactions more directly and generally produce a system of equations or a model that helps understand the nature of the relationships among the attributes. On the other hand, all of these methods do not use the class information which limit their usefulness for classification tasks. We also notice that most of them can only handle continuous attributes.

This review of literature reveals the lack of supervised constructive induction techniques that emphasize on attribute interactions, in particular the interactions involving numerical attributes, that can be used with any standard learning systems. This observation guided us during the development of the GLOREF approach described in Chapters 4 and 5. Finally, we also noticed that most methods described fail to include the user into the analysis. This prevents them from making use of additional information that the user could provide and also limit the quality of the experience for the user. The technique of relevance graphs, also introduced in Chapter 4, addresses this need by including the user in the analysis of interactions and, optionally, during the search for transformations.

Chapter 3

Empirical evaluation of the usefulness of globally relevant features

3.1 Introduction

This chapter proposes a study to assess the potential improvements that could be achieved by augmenting the initial representation with globally relevant features. The study considers an idealized situation where complete and perfect knowledge of the relationships between the initial attributes can be used to construct the new globally relevant features. The study relies on 13 learning systems to evaluate the potential improvements in accuracy due to the new features. There are three objectives:

1. Provide an estimation of the maximal improvements that could be achieved through globally relevant features.

2. Characterize various learning approaches based on their abilities to handle attribute interactions.

3. Generate *gold standard* performance results.

The first two objectives are important to justify the need for research on the problem of attribute interactions in machine learning. The third objective will help evaluate the method we propose to reduce the negative effects of attribute interactions (Chapter 6). The results from this study could also help with the algorithm selection problem. For instance, in domains where strong attribute relationships are expected, knowledge about the robustness of each approach could help in selecting the most appropriate one.

The study involves 12 artificial datasets with pre-determined attribute interactions. To maximize the applicability of the results, we constrained the experiment to relatively simple interactions that are likely to be found in real-world applications. The datasets considered

include both functional and stochastic interactions. The study also integrates a feature selection technique to help avoid difficulties related to non-relevant attributes.

This chapter has 6 additional sections. The first one explains how this study extends earlier work on the effects of attribute interactions. Section 3.3 reviews the notation of belief networks which is used in Section 3.4 to explain the data generation process. Section 3.5 presents the experimental methodology and the results obtained. Section 3.6 describes the challenges for this research. Finally, the last section summarizes this chapter.

3.2 Related Work

In the recent years, we have seen a number of research projects focusing on the problem of attribute dependencies with the naive-Bayes learning method. Most of these efforts propose extensions that may help improving the results of the naive-Bayes in presence of attribute dependencies. The resulting approaches are collectively called *semi-naive Bayesian classifiers*. The semi-naive Bayesian classifiers typically follow one of two strategies: manipulation of the data prior to the application of the naive-Bayes algorithm [Kononenko, 1991; Langley & Sage, 1994; Pazzani, 1996] or enhancement of the induction algorithm [Langley, 1993; Kohavi, 1996].

Domingos & Pazzani [1997] explains that these extensions to the naive-Bayes approach usually lead to moderate success. The paper suggests that this is due to the fact that naive-Bayes is actually more robust to attribute dependencies than is commonly thought. They support their claim through a large-scale experiment as well as a theoretical analysis of the optimality of the naive-Bayes. The goal of the experiment performed by Domingos & Pazzani [1997] was to empirically study the effects of attribute dependencies on the naive-Bayes.

To conduct their experimental study, Domingos & Pazzani [1997] compared the performance of the naive-Bayes with three other learning systems (C4.5 [Quinlan, 1993], PEBLS [Cost & Salzberg, 1993], and CN2 [Clark & Niblett, 1988]) over 28 datasets from the UCI repository [C. Blake & Merz, 1998]. They also estimated the degree of dependencies in each domain using a measure of pairwise attribute dependence [Kononenko, 1994]. They report that on average the accuracy of the naive-Bayes is significantly superior to the other systems. They also report that the naive-Bayes outperforms the other learning approaches in some domains in which the degree of attribute dependencies appears to be high. From these observations, they concluded that the naive-Bayes is not much influenced by attribute dependencies or interactions.

Our study also evaluates the effects of attribute interactions but from a different perspective. Instead of adopting a comparative approach with other learning algorithms, we control the levels of attribute interactions and observe the consequences on the performance of several learning systems. We control the attribute interactions by regulating the data generation process and by adding new features that account for these interactions. The observations obtained provide direct estimations of the impacts of attribute interactions on various learning approaches. At the same time, they indicate the magnitude of improvements that could be achieved by extending the initial representation. Such information cannot be extracted from a purely comparative study.

3.3 Graphical notation of attribute dependencies

We rely on the notation of *belief networks* [Pearl, 1988] to represent the interactions among the attributes. A belief network is a directed acyclic graph (DAG). Each node (or vertex) represents an attribute. In this chapter, we use a diamond to represent the class attribute C and circles for all other attributes. We note that the arcs are oriented. The attribute at the origin of an arc is named the *parent* while the attribute at the destination is named the *child*. The general rule for interpreting the relationships states the each attribute is said to be independent of its non-descendents given its immediate parents.

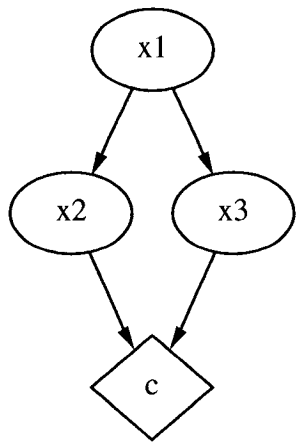


Figure 3.1: A simple Belief Network.

Figure 3.1 presents a belief networks for a simple domain with three input attributes X_1, X_2, and X_3. This graph indicates that the attributes X_2 and X_3 are independent given

the value of attribute X_1. Similarly, the target attribute C is independent of X_1 given the values of the attributes X_2 and X_3. On the other hand, the attributes X_1 and X_2 are assumed to interact and similarly for the attributes X_1 and X_3. We shall specify the nature of the interactions during the data generation process, which is explained in the next section.

3.4 Data generation

The importance and nature of attribute interactions are likely to change over different domains. To efficiently study the potential effects of attribute interactions (and the benefits of the new representation), one must try to experiment with various scenarios. Our experiment includes the six attribute interaction structures shown by the belief networks in Table 3.1. The data generation step produces two datasets for each belief network. The first dataset contains functional interactions while the second dataset contains stochastic interactions. We complete the data generation process by adding, to each dataset, new globally relevant features that eliminate the necessity of considering attribute interactions. We name these new features *theoretical features* to stress the fact that they are based on complete and perfect knowledge of the relationships among the initial attributes. This section is divided in three parts. The first two explain the generation of the datasets with functional and stochastic attribute interactions, respectively. The third part discusses the creation of the theoretical features.

Before discussing the data generation, let us mention that all datasets comprise continuous attributes only. This restriction will make it possible to re-use the same data in Chapter 6 to compare our solution with statistical techniques that can only handle continuous attributes. For simplicity, we assume binary classification problems where the two possible values (0 and 1) have equal probability. All datasets have 1000 instances. Throughout this section, we shall use $\mathcal{A}$ to represent the set of all initial attributes $\{X_1, X_2, \ldots, X_p\}$ for the given dataset. We use $\mathcal{C}$ to denote the set of attributes that are influenced by at least one other attribute. The attributes in $\mathcal{C}$ correspond to the child nodes in the belief network. We say that an attribute is relevant if there is a path between this attribute and the class attribute. We use $\mathcal{R}$ to represent the set of relevant attributes.

We observe that almost all relevant features in the belief networks considered (Table 3.1) are also child attributes. Moreover, except for the network N2, all child attributes have at least one non-relevant attribute for parent. As we shall discuss while we introduce the interactions between the attributes in the next two sections, these characteristics are

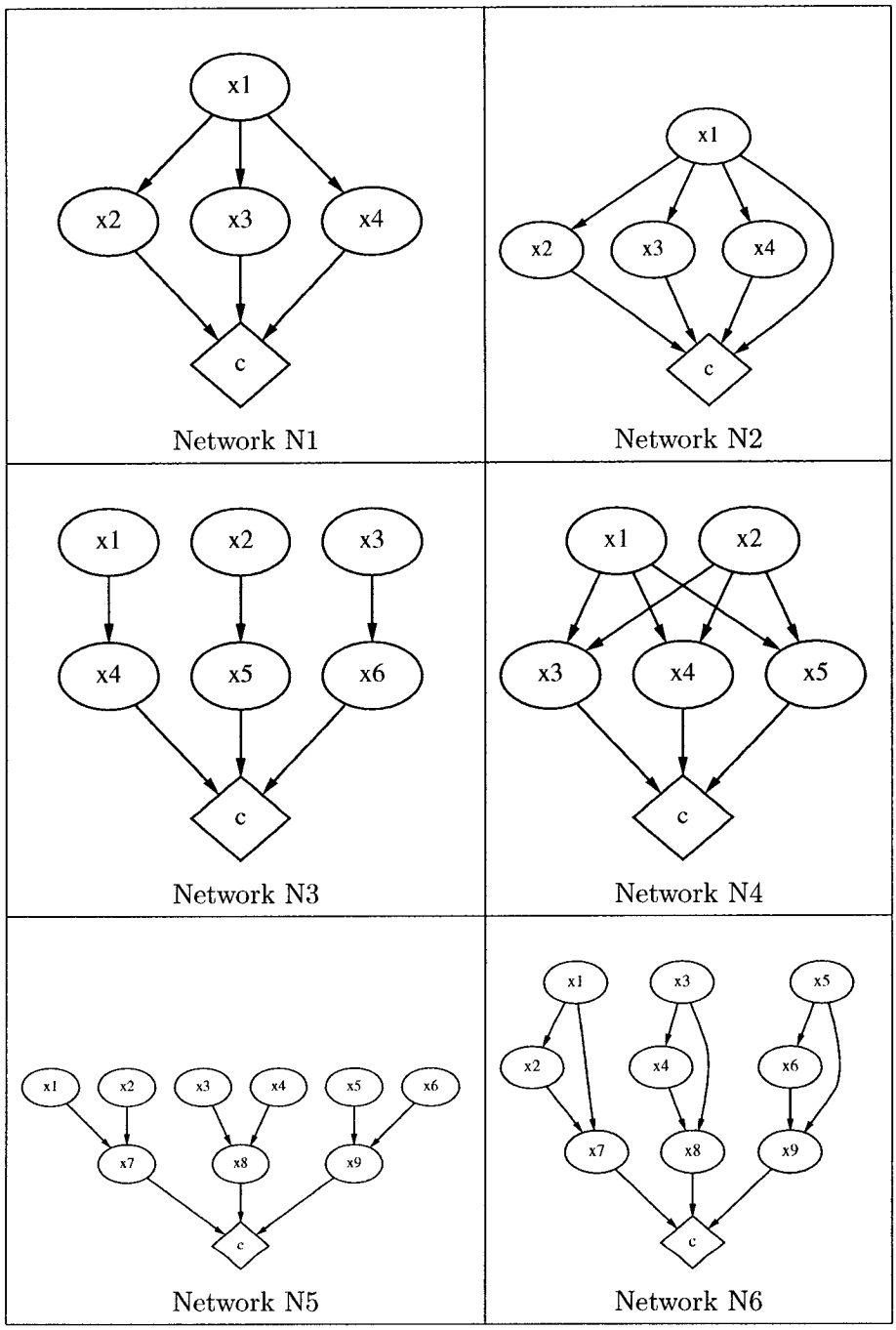


Table 3.1: Graphical representations of the interactions in the synthetic datasets.

required to ensure that the initial attributes are not already highly globally relevant.

3.4.1 Functional attribute interactions

This section describes the generation of the dataset for a given belief network where the arcs in the network represent *functional* dependencies. The presentation assumes that the class values have been generated in a pre-processing step. The data generation process starts by creating the relevant attributes. Precisely, each attribute $X_i \in \mathcal{R}$ is defined as:

$$X_i \sim \begin{cases} \mathrm{N}(\mu_{i0}, \sigma_{i0}) & \text{if } C = 0 \\ \mathrm{N}(\mu_{i1}, \sigma_{i1}) & \text{if } C = 1 \end{cases} \tag{3.1}$$

where

$\mathrm{N}(\mu, \sigma)$ refers to the normal distribution with mean μ and variance σ;

C is the class value;

μ_{i0} is random number between 0 and 1000 to represent the mean of X_i for all observations with class value equals 0 ;

σ_{i0} is a random number between 0 and $10 * \mu_{i0}$ to represent the variance of X_i for all observations with class value equals 0 ;

$\mu_{i1} = \mu_{i0} + a\sqrt{\sigma_{i0}}$, with a a random number between -2 and 2, is the mean of X_i for all observations with class value equals 1;

σ_{i1} is a random number between 0 and $10 * \mu_{i1}$ to represent the variance of X_i for all observations with class value equals 1.

As illustrated in Figure 3.2, the constraint on μ_{i1} ensures that the distance between the means of the negative and positive instances (class values 0 and 1, respectively) for a given attribute X_i is never greater than two times the standard deviation of X_i over the negative instances. This limits the relevance of the attributes by forcing some overlap over the two distributions. Without such a constraint, the attributes could be so relevant that the learning would be trivial (and therefore not realistic).

The second step generates the values for the remaining attributes (i.e., all attributes $X_i \in \mathcal{A} \setminus \mathcal{R}$). These attributes are not linked to the class attribute. Accordingly, we specify

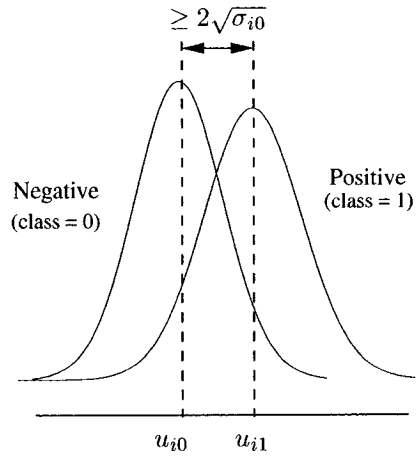


Figure 3.2: Illustration of the parameters controlling the relevance on the initial attributes.

the distributions of these attributes without using the class attribute. Precisely, we define each attribute $X_i \in \mathcal{A} \setminus \mathcal{R}$ as:

$$X_i \sim \mathrm{N}(\mu_i, \sigma_i) \tag{3.2}$$

where μ_i is a random number between 0 and 1000 and σ_i is a random number between 0 and $10 * \mu_i$.

The last step introduces the functional interactions between the child nodes and their parents in the given belief network. We introduce these interactions by updating each attribute $X_i \in \mathcal{C}$ as follow:

$$X_i = X_i + \sum_{X_j \in \mathrm{Parents}(X_i)} w_{ij} X_j \tag{3.3}$$

where Parents(X_i) represents the set of parent(s) of attribute X_i in the given network and w_{ij} a random number indicating the strength of the interactions between the attributes X_i and X_j. This formula combines the initial values for X_i with the ones from its parents in a linear manner, producing linear interactions. To avoid potential issues with recursivity, it is important to start by updating the child nodes that are the immediate descendants of the roots in the graphs and then follow the direction of the arcs.

As mentioned earlier, the graphs have been designed such that almost all attributes in $\mathcal{R}$ are also included in $\mathcal{C}$. In turns, the attributes in $\mathcal{C}$ tend to have at least one non-relevant parent. This means that Equation (3.3) affects almost all relevant attributes computed with

Equation (3.1) by forcing them to interact with one or more non-relevant attributes generated by Equation (3.2). This process systematically introduces interactions that increase the dispersion of the instances and reduce the univariate relevance of the attributes. We expect the learning algorithms that are based on the assumption on weak interactions to have difficulties achieving high accuracy with such attributes.

It is clear that many other types of attribute interactions could have been introduced. For instance, one might also consider higher order polynomials and non-linear constructs involving exponential or logarithmic functions. We limit the experiments to the simplest form of interactions because they are predominant in real world applications. We also want to keep a conservative approach to evaluate the potential benefits of a new data representation by keeping the initial attribute relationships as simple as possible.

3.4.2 Stochastic interactions

This section describes the generation of the dataset for a given belief network where the arcs in the network represent *stochastic* interactions. Again, the presentation assumes that the vector of class values, noted C, is readily available. We introduce the stochastic dependencies through a multivariate distribution for which we can parameterize the dependencies. In particular, we focus on the multivariate normal distribution.

A multivariate normal distribution is completely specified by two parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. The parameter $\boldsymbol{\mu}$ represents for the mean vector defined by

$$\boldsymbol{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_p \end{bmatrix}$$

where μ_i is the mean of attribute X_i. The parameter $\boldsymbol{\Sigma}$ is the *variance-covariance* matrix defined as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1p} \\ \sigma_{21} & \sigma_{22} & \dots & \sigma_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{p1} & \sigma_{p2} & \dots & \sigma_{pp} \end{bmatrix}$$

where σ_{ii} is the variance of X_i and σ_{ij} represents the covariance between attributes X_i and X_j. The covariance σ_{ij} indicates the association between attributes X_i and X_j. The variance-covariance matrix is a symmetric matrix (i.e., $\sigma_{ij} = \sigma_{ji}$ for all i and j).

In addition to $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, it is often useful to consider the *correlation matrix* $\mathbf{R}$ defined as

$$\mathbf{R} = \begin{bmatrix} 1 & r_{12} & \dots & r_{1p} \\ r_{21} & 1 & \dots & r_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & \dots & 1 \end{bmatrix}$$

where each *coefficient of correlation* r_{ij} is defined as

$$r_{ij} = r_{ji} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}}\sqrt{\sigma_{jj}}}$$

Each coefficient of correlation r_{ij} measures the strength of the linear association between the attributes X_i and X_j. The coefficient of correlations are usually easier to interpret than the covariance because they are bound between -1 and 1. When $r_{ij} = 0$, we say that there is a lack of linear association between the attributes X_i and X_j. When it approaches 1 or -1, then we say that there is evidence of linear interactions.

To generate the desired data from the multivariate normal distribution, we proceed as follows. First, we randomly generate the variances $\sigma_{11}, \dots, \sigma_{pp}$ and two mean vectors $\boldsymbol{\mu}_0 = [\mu_{10}, \mu_{20}, \dots, \mu_{p0}]$ and $\boldsymbol{\mu}_1 = [\mu_{11}, \mu_{21}, \dots, \mu_{p1}]$, where $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$ are for the first and second classes, respectively. We use the same constraints for σ_{ij} and μ_{ij} as with the previous approach (Equation (3.2)). Then, we generate the correlation matrix $\mathbf{R}$ by assigning a random real number to each r_{ij} with the following constraints:

1. $|r_{ij}| \in]0.5, 1.0]$ if and only if attributes X_i and X_j are conditionally dependent according to the given belief network (i.e., there is an arc between X_i and X_j in the belief network).

2. $r_{ij} \in [-0.2, 0.2]$, otherwise.

These constraints ensure a relatively high coefficient of correlation between interacting attributes and a low coefficient of correlation between independent ones.

From $\mathbf{R}$, we compute the corresponding covariance matrix

$$\mathbf{\Sigma} = \mathbf{D}^{\frac{1}{2}}\mathbf{R}\mathbf{D}^{\frac{1}{2}}$$

where

$$\mathbf{D}^{\frac{1}{2}} = \begin{bmatrix} \sqrt{\sigma_{11}} & 0 & \dots & 0 \\ 0 & \sqrt{\sigma_{22}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sqrt{\sigma_{pp}} \end{bmatrix}$$

One additional constraint must be satisfied. To be valid the covariance matrix needs to be *positive definite*. Without that condition, one would not be able to find the roots of the covariance matrix, which are required to generate the multivariate normal data. Our approach is to randomly generate a new correlation matrix until we get a valid covariance matrix. Finally, we apply the algorithm shown in Figure 3.3 to generate the multivariate normal data for the given variance-covariance matrix $\mathbf{\Sigma}$ and mean vectors $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$. As shown in this figure, we use the Choleski factorization [Johnson, 1987] to introduce the covariance effects. The last loop accounts for the effects of the mean vectors. The algorithm `GenerateMVN` returns the dataset with the specified interactions.

3.4.3 Generation of the theoretical features

The previous two sections described the generation of datasets with attribute interactions. This section presents the generation of globally relevant features that account for the initial attribute interactions. We assume an ideal situation in which perfect knowledge of the relationships among the attributes is available. We use this perfect knowledge to create the new features, which we named the *theoretical features*. The theoretical features are computed for each dataset. The process requires one technique for the functional interactions and another one for the stochastic interactions.

The attribute interactions in a given dataset are specified by the arcs entering the child nodes in the corresponding belief network. Also, let us recall that the set of attributes for these child nodes is noted $\mathcal{C}$. To eliminate the necessity of taking into account the initial relationships, we define, for each X_i in $\mathcal{C}$, a new feature $Theo\_X_i$ that encodes the joint information from X_i and its parent(s). For example, let us consider the first belief network for this experiment, which is represented by network N1 in Table 3.1. There are three child nodes in this network and the corresponding attributes are X_2, X_3, and X_4. These three

Algorithm GenerateMVN

Input: Two $p \times 1$ mean vectors $\boldsymbol{\mu}_0$ and $\boldsymbol{\mu}_1$, a $p \times p$ variance-covariance $\boldsymbol{\Sigma}$, and a 1000×1 class vector C.

Output: A multivariate normal datasets $\boldsymbol{X}_{1000 \times p}$.

For $i = 1$ to 1000
  For $j = 1$ to p
    $\boldsymbol{Z}[i,j] \leftarrow \mathrm{N}(0,1)$ /* A normal value between 0 and 1 */
$\boldsymbol{L} \leftarrow$ cholesky_root($\boldsymbol{\Sigma}$)
$\boldsymbol{X} \leftarrow \boldsymbol{LZ}$
For $i = 1$ to 1000
  For $j = 1$ to p
    If $C[i] = 0$
      $\boldsymbol{X}[i,j] \leftarrow \boldsymbol{X}[i,j] + \boldsymbol{\mu}_0[j]$
    Else
      $\boldsymbol{X}[i,j] \leftarrow \boldsymbol{X}[i,j] + \boldsymbol{\mu}_1[j]$
return $\boldsymbol{X}$

Figure 3.3: Algorithm `GenerateMVN` generates a multivariate normal dataset given a covariance matrix and a mean vector for each class.

attributes interact with X_1. As shown in Figure 3.4, the new feature $Theo\_X_2$ eliminates the need to take into account the interaction between X_2 and X_1 by combining the information from these two initial attributes. Similarly, the features $Theo\_X_3$ and $Theo\_X_4$ account for the interactions between X_3 and X_1, and between X_4 and X_1, respectively. As we shall see below, the theoretical features combine the initial attributes in such a way as to maximize their global relevance.

In the case of functional attribute interactions, the global relevances of the initial attributes were limited by the application of Equation (3.3). To maximize the global relevance, we need to reverse the effects of this equation. We achieve this by creating a new feature

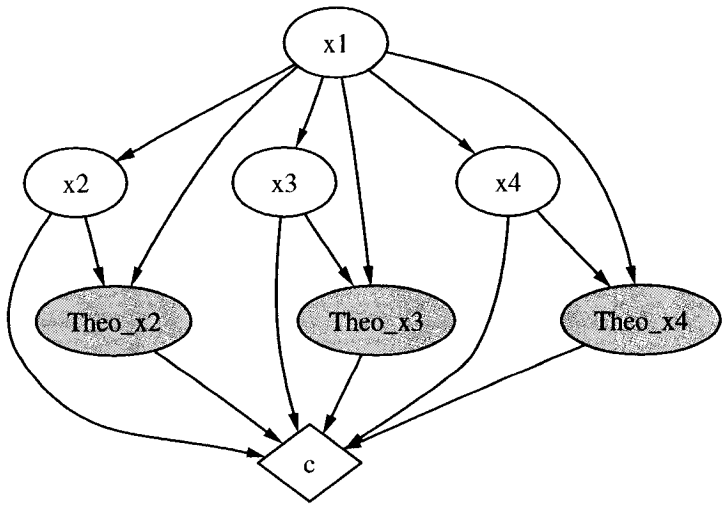


Figure 3.4: The belief network N1 augmented with the theoretical features. There is one theoretical feature for each child node in the initial network.

$Theo\_X_i$ for each $X_i \in \mathcal{C}$ such that

$$Theo\_X_i = X_i - \sum_{X_j \in \text{Parents}(X_i)} w_{ij} X_j \tag{3.4}$$

where Parents(X_i) is the set of parent(s) of attribute X_i and w_{ij} is the weight used to represent the strength of the relation between X_i and X_j for the given dataset (Equation (3.3)).

In the case of stochastic dependencies, the attribute interactions have been introduced with the following operation in the algorithm `GenerateMVN` (Figure 3.3):

$$\boldsymbol{X} \leftarrow \boldsymbol{LZ} \tag{3.5}$$

The other operations in the algorithm further transformed $\boldsymbol{X}$ to take into account the mean values. To optimize the global relevance, we need to remove the interactions caused by the pre-multiplication with $\boldsymbol{L}$. Let us assume that $\boldsymbol{X}$ contains the final data returned by the algorithm `GenerateMVN`. We create a new data matrix $\boldsymbol{Theo\_X}$ as

$$\boldsymbol{Theo\_X} = \boldsymbol{L}^{-1}\boldsymbol{X}' \tag{3.6}$$

where $\boldsymbol{L}^{-1}$ is the inverse of $\boldsymbol{L}$ in Equation (3.5). The resulting data matrix $\boldsymbol{Theo\_X}$

contains p columns named $Theo\_X_1, Theo\_X_2, \ldots, Theo\_X_p$. These columns correspond to the initial attributes $X_1, X_2, \ldots, X_p$ with the exception that they are uncorrelated (i.e., all coefficients of covariance equal 0). As explained above, we only need to keep the new features that correspond to the child attributes in the initial network. Therefore, we keep the feature $Theo\_X_i$ if and only if $X_i \in \mathcal{C}$. The others features are disregarded. We note that this last step ensures that we have the same number of new features for both datasets (functional and stochastic) generated for a given belief network.

3.5 Experimentation

This section has two sub-sections. The first one describes the methodology used while the second one discusses the results of the study.

3.5.1 Methodology

We assess the usefulness of the theoretical features by comparing the performances with the initial attributes only and with both the initial attributes and the theoretical features. We use $\mathcal{A}$ to represent the initial set of attributes and $\mathcal{F}$ to represent the augmented set which also includes the theoretical features.

The presence of irrelevant attributes or features may deteriorate the performance of learning algorithms. Since the datasets generated contain irrelevant attributes, we decided to try to minimize the potential effects of irrelevant attributes by incorporating a feature subset selection technique in the experiment. We apply the feature selection technique to select a subset of $\mathcal{A}$, noted FS($\mathcal{A}$), and a subset of $\mathcal{F}$, noted FS($\mathcal{F}$). For each dataset and classifier system, we evaluate the expected accuracy of models learned using the full and reduced feature sets (i.e., $\mathcal{A}$, FS($\mathcal{A}$), $\mathcal{F}$, and FS($\mathcal{F}$)).

We evaluate the accuracies through a 10-fold cross-validation procedure. We repeat the procedure for all combinations of datasets and learning systems. Since all artificial datasets have 1000 instances, the training dataset in any given fold has 900 instances and the remaining 100 instances are kept for testing. In each fold, the procedure performs the following tasks:

Feature selection We use feature selection to produce the two feature subsets FS($\mathcal{A}$) and FS($\mathcal{F}$). This step uses 30% of the training datasets for the current fold[1].

[1]This means that the feature selection step uses 270 instances ($.3 * 900 = 270$). We did not utilize the full training dataset during feature selection for consistency with the other experiments described in Section 6.2.

Learning the models We run the given learning system four times, varying the input attributes as follow:

1. All initial attributes ($\mathcal{A}$).
2. Subset of initial attributes (FS($\mathcal{A}$)).
3. All initial attributes and all theoretical features ($\mathcal{F}$).
4. Using subset of all initial attributes and all theoretical features (FS($\mathcal{F}$)).

This step produces four models; one for each feature set. All the training data for the current fold is used to learn these models.

Evaluation of the models We evaluate the accuracies of the four models generated above on the test data for the current fold.

Upon completion of the 10^{th} fold, we obtain the final estimations of accuracy by averaging the results from the 10 folds. We obtain four accuracy estimations; one for each set of models. In addition to the average accuracies, we also record the corresponding standard deviations.

We use the same feature selection method across all experiments. It is a filter method based on ID3 and a forward search [Liu & Motoda, 1998]. The method works as follow. The search starts with an empty feature subset, then it builds an ID3 model for each feature (or attribute) independently. The feature associated with the most accurate model is added to the selected feature subset. To select the second feature, the search considers all models using the first selected feature and one of the remaining features. The remaining feature that maximally increases the accuracy, over the previous best model, is added to the selected feature subset. The process stops when no feature further increases the accuracy of the best model obtained or when all features have been added to the set of selected features.

We applied this procedure to the 12 artificial datasets considered and 13 learning systems described in Appendix A. All learning systems have been used with their default settings. The next section discusses the results obtained.

3.5.2 Results

Tables B.1 to B.12 in Appendix B provide the detailed results of the experiments. There is one table for each dataset. We named the artificial datasets using four characters. The first two represent the corresponding belief network in Table 3.1 and the last two indicate the type of the interactions among the attributes. We use the letters 'F1' for functional

attribute interactions and 'MN' for the multivariate normal stochastic interactions. For example, the dataset 'N1F1' indicates that the structure of the interaction is given by the belief network named N1 and that the arcs in the network represent functional interactions.

Each row in Tables B.1 to B.12 displays the results for a given classification system. There are six columns. The first one gives the name of the classification algorithm considered. The four following columns present the average accuracy and standard deviations for the four models learned. In particular, the first two columns present the results for models learned from all initial attributes and from a subset of the initial attributes, respectively. The other two columns present the average accuracies and standard deviations when the learning systems also use the theoretical features, without and with feature selection, respectively. Finally, the last column displays the average difference in accuracy between the best model using the theoretical features and the best model using the initial attributes only for the current classifier. The best model with the initial attributes is the one with maximal accuracy between the run involving all initial attributes and the run involving a subset of them only. Similarly, the best model with the theoretical features is the one with maximal accuracy between the two runs involving the theoretical features (with and without feature selection) for the current classifier. A positive difference indicates that the models using the theoretical features perform better for the given classifier, and vice versa. A star besides the difference denotes significance at the 0.05 level using the standard t-test to compare group means.

Figure 3.5 offers a quick view of all the results combined. There is one point for each combination of datasets and classifiers for a total of 156 points (12 datasets * 13 classifiers). The horizontal position represents the accuracy of the best model based on the initial attributes only while the vertical position denotes the accuracy of the best model that also involves the theoretical features. The points located above the diagonal line indicate that the theoretical features have had a positive effect on accuracy. All points below the diagonal line represent experiments in which the use of the theoretical features resulted in a decrease in accuracy. Points on the diagonal line stands for the experiments with identical accuracy between models with and without the theoretical features.

We notice that almost all the points are located above the line. This clearly demonstrates the overall positive effect of the theoretical features on the accuracy. We also observe that the few points located below the reference line are actually very close to it, which means that using the theoretical features never caused an important decrease in accuracy. On the other hand, the importance of the positive effects of the theoretical features is much greater since we find many points located far above the reference diagonal line. We now specialize

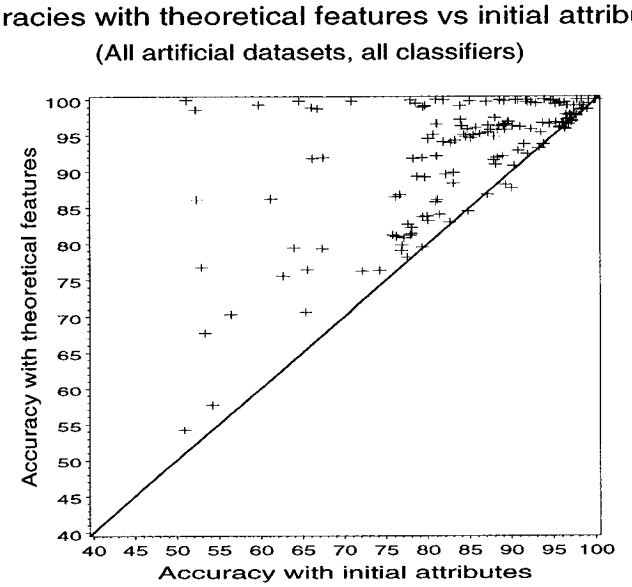


Figure 3.5: Accuracies using theoretical features and initial attributes versus accuracies with initial attributes only.

the analysis of the results by dataset and by classifier.

Table 3.2 summarizes the results by dataset. Each row summarizes the results from all classification systems for the dataset specified in the first column. The following four columns report statistics on the difference between the best theoretical model and the best model with the initial attributes only. These statistics are computed from the values in the last column of Tables B.1 to B.12. The next three columns report the number of times that the theoretical features increased accuracy, decreased accuracy, and did not change the accuracy, respectively. The sum of these three columns is 13 since we used 13 classification algorithms. The last two columns provide the number of times we obtained a statistically significantly better accuracy and a statistically significantly worse accuracy, respectively (at level 0.05).

The overall average of accuracy improvements due to the use of the new features is 8.01. As indicated by the last two lines in Table 3.2, the average improvement with the datasets having stochastic interactions is 9.43 and the average improvement with the datasets having functional interactions is 7.20. The larger gain for stochatic interactions may be partially explained by the two datasets for the second belief network which lead to improvements of .13 (N2F1) and 12.9 (N2MN). This dataset is the only one which has a relevant attribute that is not affect by any other attributes (this is attribute X_1). As explained in Section 3.4,

Theoretical vs initial features by dataset

Dataset	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
N1F1	4.05	4.32	0.40	13.9	13	0	0	6	0
N1MN	3.75	2.67	0.20	10.9	13	0	0	5	0
N2F1	0.13	0.50	-.50	1.40	5	5	3	0	0
N2MN	12.9	8.49	-2.3	25.7	11	2	0	11	0
N3F1	1.27	1.84	-.30	6.90	12	1	0	2	0
N3MN	3.68	3.34	-.30	12.9	11	2	0	7	0
N4F1	11.3	10.6	1.80	39.6	13	0	0	13	0
N4MN	14.4	14.9	0.10	46.4	13	0	0	12	0
N5F1	7.75	5.70	2.10	25.1	13	0	0	12	0
N5MN	16.3	14.1	0.00	48.8	12	0	1	12	0
N6F1	11.6	7.36	3.00	33.8	13	0	0	13	0
N6MN	9.05	4.59	0.50	15.5	13	0	0	11	0
All	8.01	9.22	-2.3	48.8	142	10	4	104	0
Functional	7.20	7.62	-.30	39.60	69	6	3	46	0
Stochastic	9.43	10.67	-.30	48.8	73	4	1	58	0

Table 3.2: Comparison, by dataset, of accuracies between models with theoretical features and models with initial features only.

because of this characteristic, the initial attribute X_1 may have been already highly globally relevant. This would explain why the the addition of other globally relevant features did not produce any advantage for dataset N2F1. This explanation is not inconsistent with the result for N2MN since the relevance of X_1 in N2MN might be different than the relevance of X_1 in N2F1. Also worth noting, is the fact that the stochastic datasets appear to represent harder problems than the functional ones since the initial accuracies with the stochastic datasets were systematically lower than the accuracies obtained with the datasets having functional interactions. This is verified by looking at the accuracy estimations in Tables B.1 to B.12.

If we make abstraction of N2MN, it becomes clear that the improvements with the last three dependency structures (N4, N5, and N6) tend to be higher than with the first three structures (N1, N2, and N3). This result was expected since the interactions are

slightly more complex with the last three networks than with the first three networks. The standard deviations are relatively high in comparison with the average improvements. This indicates that the usefulness of the new features is also a function of the classification system employed. We will further investigate this aspect with Table 3.3, which summarizes the results by classification system.

The 'Min' and 'Max' columns in Table 3.2 clearly indicate that adding the theoretical features can lead to great improvements in accuracy (up to 48.8 for the datasets considered) while the potential negative effects on accuracy are negligible (no more than 0.5 decrease in accuracy for the datasets considered). The potential benefits are further illustrated by the last five columns in Table 3.2. Among the 156 cross-validation experiments performed, the theoretical features helped increase the accuracy in 142 cases ($\approx$ 91.0%), reduced the accuracy in 10 cases ($\approx$0.06%), and did not affect the accuracy in 4 experiments ($\approx$0.03%). When considering only the statistically significant differences, we find that the theoretical features helped increase the accuracy in 104 cases ($\approx$66.7%) while never reducing it significantly.

Table 3.3 stratifies the results by classification system instead of by dataset. Each row summarizes the results from all datasets for a given learning system. The statistics computed to summarize the results are the same as with Table 3.2. We observe that the theoretical features greatly helped with the following classifiers: HyperPipes (21.7), OneR (14.1), DecisionTable (10.4), and NaiveBayes (10.0). All of these classifiers are very limited with respect to attribute interactions, it is therefore not surprising to see that their performance greatly increase when we provide them with globally relevant features. We also note that the theoretical features boosted the accuracies of all other classification systems. In particular, the number of datasets with statistically significant increase in performance varies from 3 out of 12 (25%) to 11 out of 12 ($\approx$91.7%). These results show that even the most sophisticated learning systems are likely to benefits from highly globally relevant features.

3.6 The challenges of this research

The results of the study performed in this chapter suggest that performance of many machine learning algorithms could be improved by adding globally relevant features to datasets composed of low level interacting attributes. Actually, the study shows that significant improvements are expected even with learning algorithms that do not assume attribute independence. Moreover, there is a great potential applicability for such a technique since

Theoretical vs initial features by classifier

Dataset	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
BaggingDT	3.20	3.39	-1.1	8.20	10	2	0	6	0
BoostingDT	2.28	2.71	-2.3	7.10	10	2	0	5	0
DecisionStump	12.9	11.1	0.00	32.9	11	0	1	11	0
DecisionTable	10.4	7.47	-.50	22.0	11	1	0	9	0
HyperPipes	21.7	17.3	0.50	48.8	12	0	0	11	0
IB1	5.22	4.59	-.20	11.3	11	1	0	7	0
IB5	4.47	3.90	0.00	12.3	11	0	1	7	0
J48	6.74	4.11	-.20	12.7	11	1	0	10	0
KernelDensity	4.79	4.14	-.20	10.3	11	1	0	7	0
NaiveBayes	10.0	6.48	0.20	19.1	12	0	0	9	0
OneR	14.1	11.9	0.00	35.3	11	0	1	10	0
PART	6.42	4.10	-.30	12.3	11	1	0	9	0
SMO	1.93	3.15	-.30	11.2	10	1	1	3	0
All	8.01	9.22	-2.3	48.8	142	10	4	104	0

Table 3.3: Comparison, by classifier, of accuracies between models with theoretical features and models with initial features only.

low level interacting attributes characterize many real world datasets [Rendell & Seshu, 1990; Langley & Simon, 1995]. As discussed in Section 2.3.1, there are a number of areas of research that propose techniques to enhance the initial data representation. However, none of the existing techniques can generate continuous features that are optimized for learning tasks.

It is important to mention that the technique used in this chapter to generate the globally relevant features cannot be considered as a potential solution since it relies on the unrealistic assumption of perfect and complete knowledge of the interactions among the initial attributes. Now that we have a better appreciation of the potential benefits of globally relevant features, we need to conceive, implement, and evaluate an approach that can generate such features for datasets for which we may not have any prior information about attribute interactions.

3.7 Summary

This chapter presented an empirical study to evaluate the potential improvements that could be achieved by augmenting the initial representation with globally relevant features. We first explained how this study complements earlier work on the analysis of the effects of attribute interactions. The study, which is entirely based on synthetic datasets, proposed a direct mean to evaluate the effect of various types of attributes interactions. The automatic generation of datasets with pre-determined attribute interaction characteristics is a difficult problem. We provided two general techniques to solve this problem. One of them generates datasets having functional interactions among the attributes while the other one generates datasets with stochastic interactions. We used the standard notation of belief networks to specify the interaction structures for the various domains considered. We described the computations of the new globally relevant features. We refer to these as theoretical features since their computations relied on perfect knowledge of the initial attribute interactions. We described the methodology followed to evaluate the expected effects of the new globally relevant features and then discussed the results. These results clearly demonstrated that globally relevant features can significantly improve the accuracy with all of the 13 learning systems considered. The results also indicated that the use of highly globally relevant features is very unlikely to reduce the accuracy. Finally, the last section explained that the main challenge for the other parts of this dissertation is to develop a general pre-processing approach to generate globally relevant continuous features. The next two chapters address this challenge.

Chapter 4

Analysis and visualization of the impacts of attribute interactions on relevance

4.1 Overview of the GLOREF approach

As explained in the introduction, the objective of this research is to enhance the performance of learning systems in domains with attribute dependencies. The approach we proposed is to improve the data representation by generating globally relevant features. We named this approach GLOREF (GLObally REelevant Features).

GLOREF works as a pre-processor and can be used with any standard learning algorithm. This is important as the problem of attribute dependencies affects several learning paradigms (see chapter 3). The approach can make use of available background information about possible attribute dependencies but does not require such information. The input is the initial dataset which contains at least one numerical attribute. The output is an augmented dataset composed of the initial attributes plus zero or more new numerical features.

As shown in Figure 4.1, our approach to generate globally relevant features has two phases: the analysis of relevance and the feature generation. The first phase computes information to characterize the interactions among the attributes in terms of their impact on learning. This information is stored in data structures named *relevance matrices*. The second phase uses the relevance matrices to search for transformations that will produce new useful features. These features are computed and added to the initial data. Finally, the usual learning steps such as feature selection and learning are applied to induce the classification models. We use two chapters to describe the GLOREF approach; this chapter focuses on the analysis of relevance and the next chapter presents the feature generation phase.

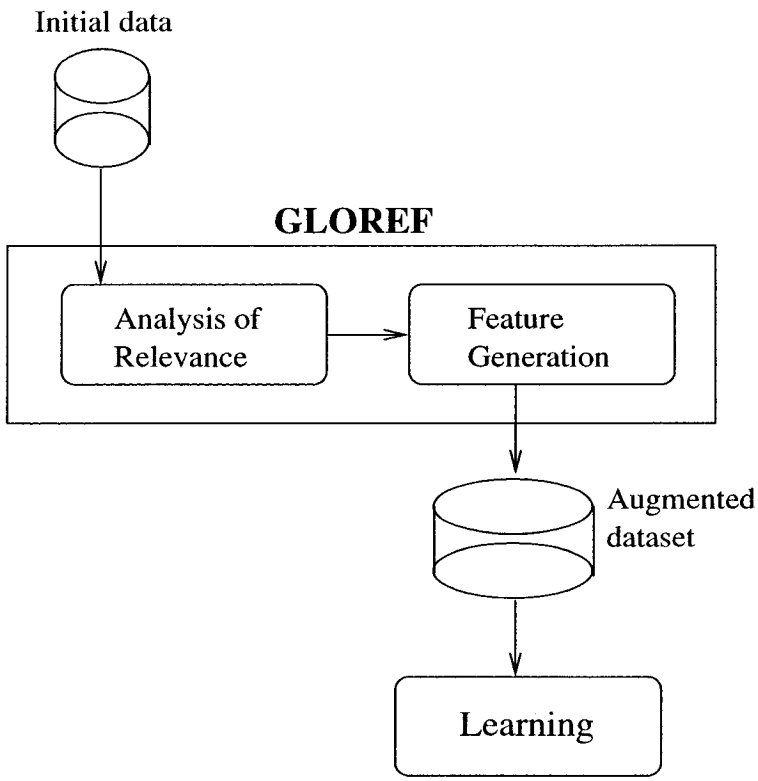


Figure 4.1: Overview of the GLOREF approach for feature generation.

As part of the analysis of relevance, we introduce a new visualization technique called *relevance graphs*. These graphs efficiently summarize all important information contained in the relevance matrices. This helps during exploratory analysis to better understand the nature and strength of the interactions among the attributes. With this understanding, it becomes easier to develop adequate transformations to generate the new features.

As we will see in the next chapter, the GLOREF approach is completely automated in the sense that it performs all the steps from the identification of important interactions to the generation of the new features without requiring human intervention. The relevance graphs complement this process by providing an efficient way for the researcher or analyst to visualize the numerical information computed during the analysis of interactions. They help understand how the attribute interactions observed affect the complexity of the learning task. Moreover, they can be used to extend this research or develop alternative solutions. The second part of this chapter details the relevance graphs and illustrates their use in the analysis of attribute interactions.

There are four additional sections in this chapter. The first one describes the analysis of relevance along with the computation of the relevance matrices. Section 4.3 introduces the relevance graphs. Section 4.4 presents the interactive relevance graphs system we have developed to allow the researcher to investigate various transformations interactively. The last section summaries this chapter.

4.2 Analysis of relevance

Most real applications are characterized by low level attributes with important interactions among them [Pérez & Rendell, 1996a]. However, not all of these interactions are necessarily detrimental to learning. To derive useful features, we first need to identify the interactions that are going to interfere with the learning algorithms. We perform this by evaluating the attribute interactions in terms of their effects on the relevance of the attributes involved.

In this chapter, we limit the discussion to the analysis of interactions between two attributes. One is named the *response* attribute and the other is the *explanatory* attribute. Explanatory attributes can be numeric or nominal while response attributes have to be continuous. For each pair (response-explanatory), the analysis of relevance computes the effect of the explanatory attribute on the relevance of the response attribute. The process has two steps: data partitioning and computation of relevance matrices. These two steps are described in Sections 4.2.3 and 4.2.4. However, before explaining these steps, we first introduce the notation used and discuss the importance of attribute relevance in the analysis of interactions.

4.2.1 Notation

Consider a training data $S = \{s_1, s_2, \ldots, s_N\}$ composed of N instances. Each instance s_i is described by a finite set of variables or attributes $\{X_1, X_2, \ldots, X_p, Y_1, Y_2, \ldots, Y_q, C\}$. We use capital letters, such as X, Y, for attribute names and lowercase letters x, y for specific values taken by these variables. Unless noted otherwise, we use the attribute names X_i for nominal attributes and Y_i for numerical attributes. The attribute C is the class attribute. The domain of a nominal attribute X is noted $\mathrm{Dom}(X) = \{X(1), X(2), \ldots, X(n)\}$ where $X(i)$ denotes the i^{th} value of attribute X. The cardinality of the $\mathrm{Dom}(X)$ is $|\mathrm{Dom}(X)|$. The value taken by attribute X in instance s is $\mathrm{val}_X(s)$. Similarly, the class value for instance s is $\mathrm{val}_C(s)$. Finally, the k possible class values for C are $\{c_1, c_2, \ldots, c_k\}$.

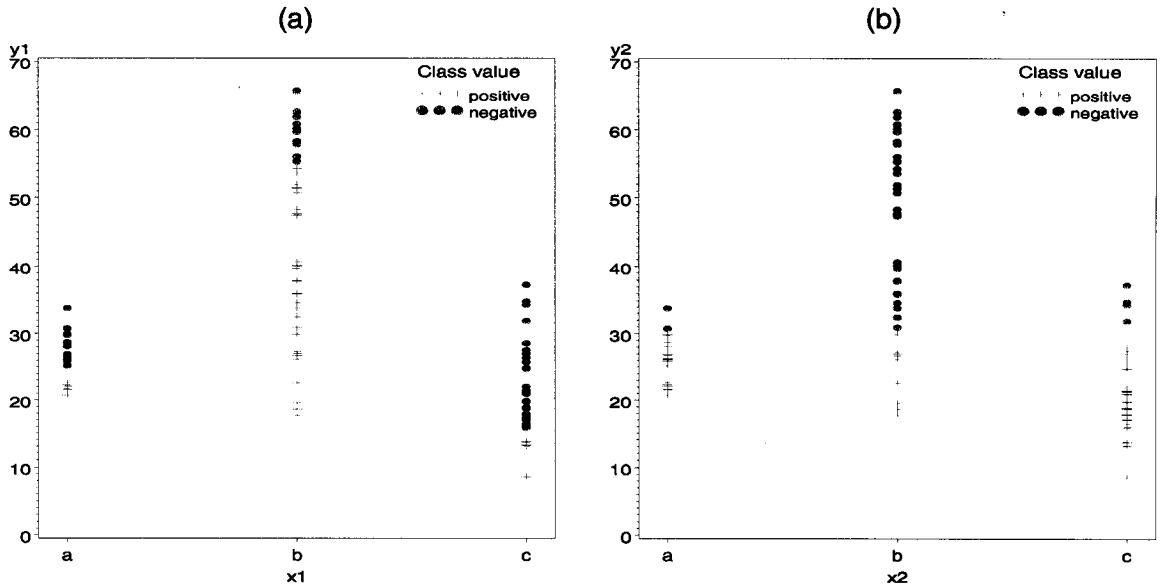


Figure 4.2: Examples of two domains with attribute interactions. The two domains are identical except for the values of the class variable.

4.2.2 The role of attribute relevance

The analysis of variable interaction is a classical topic in statistics and many techniques have been developed for this task, including: ANOVA, Regression Analysis, Analysis of Correlation, and PCA. Based on machine learning terminology, all of these techniques are called *unsupervised* since they do not make use of the class information during the analysis of the structure among the variables. Because of this particularity, these techniques cannot differentiate the attribute relationships that increase the complexity of the learning task from the ones that are harmless with respect to learning.

To illustrate the limitation of unsupervised techniques in the analysis of attribute interactions, let's consider the following example. Suppose, we have two datasets, D_1 and D_2, each having three attributes. The attributes in the first dataset are named: X_1, Y_1, and C_1. The attributes in D_2 are: X_2, Y_2, and C_2. Attributes Y_1 and Y_2 are continuous, X_1 and X_2 are nominal with the following possible values $\{a, b, c\}$, and C_1 and C_2 are the binary class variables for each domain. Moreover, suppose that the two datasets are identical except for

(a)

The ANOVA Procedure

Dependent Variable: y1

Source	DF	Anova SS	Mean Square	F Value	Pr > F
x1	2	10617.84007	5308.92004	53.10	<.0001

The ANOVA Procedure

Tukey's Studentized Range (HSD) Test for y1

Comparisons significant at the 0.05 level are indicated by ***.

x1 Comparison	Difference Between Means	Simultaneous 95% Confidence Limits		
b - a	17.272	10.806	23.738	***
b - c	22.279	16.950	27.607	***
a - b	-17.272	-23.738	-10.806	***
a - c	5.007	-1.568	11.582	
c - b	-22.279	-27.607	-16.950	***
c - a	-5.007	-11.582	1.568	

(b)

The ANOVA Procedure

Dependent Variable: y2

Source	DF	Anova SS	Mean Square	F Value	Pr > F
x2	2	10617.84007	5308.92004	53.10	<.0001

The ANOVA Procedure

Tukey's Studentized Range (HSD) Test for y2

Comparisons significant at the 0.05 level are indicated by ***.

x2 Comparison	Difference Between Means	Simultaneous 95% Confidence Limits		
b - a	17.272	10.806	23.738	***
b - c	22.279	16.950	27.607	***
a - b	-17.272	-23.738	-10.806	***
a - c	5.007	-1.568	11.582	
c - b	-22.279	-27.607	-16.950	***
c - a	-5.007	-11.582	1.568	

Figure 4.3: Output from ANOVA analysis of interaction for two domains.

the class values. Precisely, we have

$$|D_1| = |D_2| = N$$
$$\mathrm{Dom}(X_1) = \mathrm{Dom}(X_2) = \{a, b, c\}$$
$$\mathrm{Dom}(Y_1) \subseteq \mathbb{R}, \mathrm{Dom}(Y_2) \subseteq \mathbb{R},$$
$$\mathrm{Dom}(C_1) = \mathrm{Dom}(C_2) = \{0, 1\}$$
$$\mathrm{val}_{X_1}(s_i) = \mathrm{val}_{X_2}(s_j) \text{ for } i, j \in \{1, 2, \ldots, N\} \text{ and } i = j$$
$$\mathrm{val}_{Y_1}(s_i) = \mathrm{val}_{Y_2}(s_j) \text{ for } i, j \in \{1, 2, \ldots, N\} \text{ and } i = j$$

Figure 4.2 presents the scatter plots for two datasets satisfying these constraints. The left side of the figure is for the first domain and the right side is for the second domain. Each dataset has 100 instances. Different symbols are used to differentiate the negative and positive instances.

The classical statistical technique to analyze the effect of a nominal variable on a continuous variable is the ANOVA. As shown in the SAS output on Figure 4.3, the ANOVA analyses confirm that the effects of X_1 on Y_1 and X_2 on Y_2 are both significant (in both cases the p-value for the F statistic is smaller than 0.0001). Moreover, the pairwise mean comparisons reveal that the two interactions are identical (this is no surprise since the two

domains have the same values for the x and y attributes). Because they are identical, a solution (or transformation) that is appropriate for one interaction should also be adequate for the other interaction. As we shall see, this is not necessarily the case. For instance, suppose we derive a new feature in each domain respectively called Y_1' and Y_2' such that

$$Y_1' = \Gamma(X_1, Y_1; s)$$
$$Y_2' = \Gamma(X_2, Y_2; s)$$

where

$$\Gamma(X, Y; s) = \begin{cases} \text{val}_Y(s) - 25.0 & \text{if } \text{val}_X(s) = a \\ \text{val}_Y(s) - 55.0 & \text{if } \text{val}_X(s) = b \\ \text{val}_Y(s) - 15.0 & \text{if } \text{val}_X(s) = c \end{cases} \tag{4.1}$$

In this case, the resulting values for Y_1' and Y_2' would be identical since we apply the same transformation Γ and the input values for this transformation are identical in both domains.

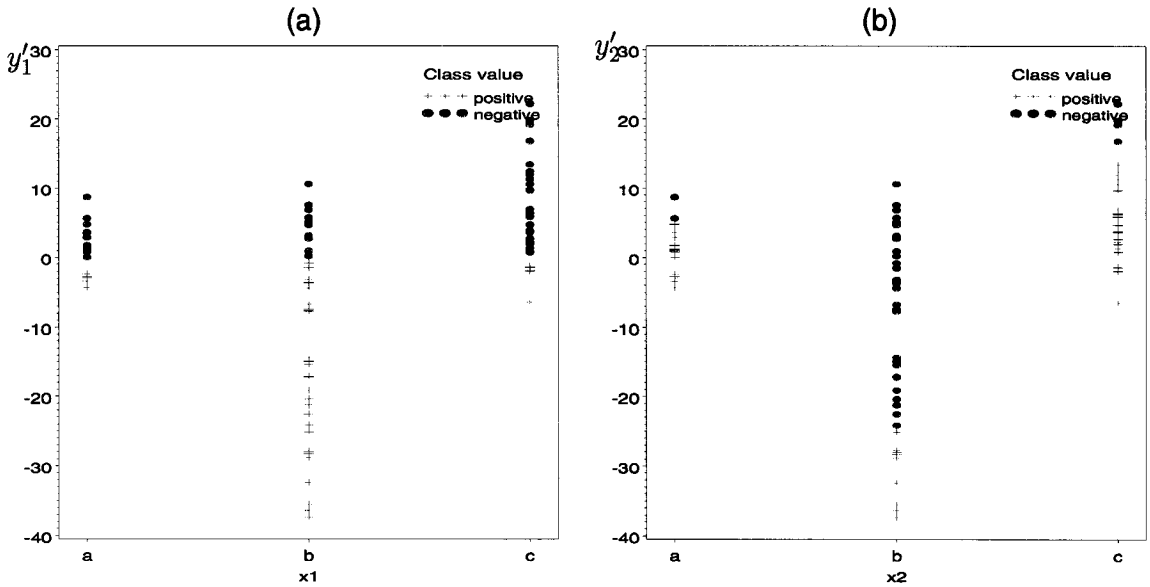


Figure 4.4: New features derived by transforming the initial Y_1 and Y_2 attributes to adjust for the effects of X_1 and X_2, respectively. We note that Y_1' facilitates the learning by comparison to Y_1 while Y_2' deteriorates the situation with respect to Y_2 (see Figure 4.2 for distribution of Y_1 and Y_2)

The scatter plots for the new features are shown in Figure 4.4. Clearly, the transformation helped with the first domain since it is easier to find a split along the Y_1' axis to correctly separate the two classes than it is with the original Y_1 axis; a split based on the test $Y_1' < 0$ perfectly separates the two classes. Any learning algorithm that assumes attribute independence is likely to achieve better performance using Y_1' than using the initial Y_1 attribute. However, the same transformation leads to an unfavorable outcome with the second domain since it is more difficult to separate the classes using Y_2' than it is using Y_2; it is impossible to build a test using only Y_2' that would split the data into 2 pure partitions while this was possible prior to transformation (using $Y_2 < 30$). To fully exploit the new feature Y_2', the learning algorithm will need to analyze it simultaneously with X_2. This is a more difficult task that exceeds the ability of several learning approaches.

This demonstrates that although the two interactions appear to be identical, they are not and each of them requires a specific solution (i.e. transformation). The ANOVA, like any unsupervised techniques, is not capable of discriminating such interactions because it ignores the class information. As demonstrated by this example, this could potentially lead to the application of transformations that are not appropriate for learning. To avoid this problem, we need a supervised technique that links the analysis of attribute interactions with the class information.

We use the concept of attribute relevance to take into account the class information. Instead of modeling the effect of the explanatory variable on the response variable, we model the effect of the explanatory attribute on the *relevance* of the response attribute. As we will see in section 4.3, this technique allows for a clear distinction between the two types of interactions present in the above datasets. As we will see in Chapter 5, the notion of attribute relevance also plays a fundamental role in guiding the search for effective transformations.

4.2.3 Data partitioning

The analysis of the effect of the explanatory attribute on the relevance of the response attribute starts by partitioning the initial dataset. We first define the notion of partition:

Definition 1 A *partition* of a dataset S is a collection $S_1, S_2, \ldots, S_m$ of non-empty subsets such that

$$S = S_1 \cup S_2 \cup \cdots \cup S_m$$

and

$$S_i \cap S_j = \emptyset$$

for all $i, j \in \{1, 2, \ldots, m\}$ and $i \neq j$. □

The strategy to partition S depends on the type of the explanatory attribute. If the explanatory attribute X is nominal, then the partition generates a subset for each value in $\mathrm{Dom}(X)$. For example, a partition of $S = \{s_1, s_2, \ldots, s_N\}$ according to a nominal explanatory attribute X with $\mathrm{Dom}(X) = \{X(1), X(2), \ldots, X(m)\}$ generates m subsets $S_1, S_2, \ldots, S_m$ where

$$S_i = \{s \in S \mid \mathrm{val}_X(s) = X(i)\} \text{ for } i \in \{1, 2, \ldots, m\} \tag{4.2}$$

When the explanatory attribute is continuous, we first *discretize* it and then partition based on the discretized values instead of the original ones. The discretization is required to limit the number of subsets.

Attribute discretization is important in machine learning since several leaning algorithms require discrete attributes. Liu et al. [2002] and Dougherty et al. [1995] provide up-to-date reviews of commonly used discretization techniques and report their performance on real world datasets. Both experiments showed that *supervised* discretization techniques tend to produce more accurate models than *unsupervised* techniques. The explanation for this superiority relates to the fact that the supervised techniques use the class information to minimize mixture of instances having different class values [Dougherty et al., 1995].

As we shall explain in the following section, during the analysis of of the effect of the interaction, we compare the relevance of the response attribute across the groups defined by the value of the explanatory attribute. To ensure that these comparisons are meaningful, we need partitions of approximately equal size and variance. Since supervised discretization techniques are not considering these constraints explicitly, they often produce groups of various sizes and variances which in turn, hurt the comparison process. To avoid this difficulty, we decided to use unsupervised discretization techniques that take into account the equal size and equal variance constraints.

Depending on the distribution of the variable, it might not be possible to obtain a single discretization schema that respects the approximately equal group size and variance criteria. Recognizing the risk of a single inadequate discretization, we decided to generate two discretized attributes for each numerical explanatory attribute. The first discretized attribute implements the equal-width intervals strategy while the second discretized attribute generates intervals with equal-frequency. Our implementation is as described in Liu et al. [2002] with the additional test ensuring that duplicated values only belong to one interval (page 401 in [Liu et al., 2002]).

Our decision to generate two discrete attributes for each numerical explanatory attribute has an impact on the performance since it increases the number of times we need to perform the analysis of relevance. Specifically, it doubles the analysis effort for each numerical explanatory attribute. The increase in computation might be problematic in applications with a large number of numerical attributes. In such domains, background information may be available to help define meaningful discretization schemas, canceling the need for automatic generation of discretized attributes.

Both discretization approaches require the user to specify the number of bins used to discretize the attributes. This choice has two consequences. First, it affects the performance since part of the analysis, such as the search for transformations, has a worst-case complexity that is exponential in the number of partitions (Section 5.2.2). Second, this choice may reduce the ability of the procedure to find any significant dependencies. In general, the fewer the intervals; the higher the likelihood of finding significant dependencies. Both of these arguments suggest that we should always target the smallest number of bins required to capture the dependency. In practice, this number is found through experimentation. As discussed in Chapter 6, we obtained good performance with 3 bins in several domains.

To treat the case of missing values, we generate an additional partition gathering all instances for which the value of the explanatory attribute is unknown. This allows the technique to detect situations where the presence of missing values is informative [Famili et al., 1997].

4.2.4 Computation of relevance matrix

As shown by the example in Section 4.2.2, to identify the attribute interactions of consequence, we need to consider the ability of the response attribute to discriminate the instances. We do this by linking the analysis of dependencies to the relevance of the response attribute. Each subset $S_1, S_2, \ldots, S_m$, generated by the partition on the explanatory attribute, is considered independently.

To analyse the relevance of the response attribute, we adopt an approach similar to the one used in the C4.5 system to deal with continuous attributes [Quinlan, 1993]. We define a cut point for each observed value of the response attribute and then evaluate the relevance, in terms of gain ratio, of binary splits on these thresholds. For example, a split

on a response attribute Y at threshold 5 of a dataset S would create two subsets

$$L = \{s \in S \mid \text{val}_Y(s) \leq 5\} \tag{4.3}$$

$$R = S \setminus L = \{s \in S \mid \text{val}_Y(s) > 5\} \tag{4.4}$$

These subsets are named the *cumulative* and *balance* sets, respectively. We add a subscript to L and R to refer to a split on a subset S_i. For example, the two subsets created by the above split on the subset S_i are

$$L_i = \{s \in S_i \mid \text{val}_Y(s) \leq 5\} \tag{4.5}$$

$$R_i = S_i \setminus L_i = \{s \in S_i \mid \text{val}_Y(s) > 5\} \tag{4.6}$$

To compute the gain ratio of a given cut point, we need to calculate, for each subset (i.e. L_i and R_i), the sum of weights of the instances from each class. The instance weight is a number between 0 and 1.0 indicating the relative importance of this instance. Supposing that the class attribute C has k values $c_1, c_2, \ldots, c_k$ and

$$L_{i,j} = \{s \in L_i \mid \text{val}_C(s) = c_j\} \text{ for } j = 1, \ldots, k \tag{4.7}$$

$$R_{i,j} = \{s \in R_i \mid \text{val}_C(s) = c_j\} \text{ for } j = 1, \ldots, k \tag{4.8}$$

$$S_{i,j} = \{s \in S_i \mid \text{val}_C(s) = c_j\} \text{ for } j = 1, \ldots, k \tag{4.9}$$

$$\text{weight}(a, S) = \text{weight of the a}^{th} \text{ instance in } S \text{ is a number in } [0, 1.0] \tag{4.10}$$

then the sum of weights for each class in the cumulative and balance sets are

$$|L_{i,j}|^{(w)} = \sum_{a=1}^{|L_{i,j}|} \text{weight}(a, L_{i,j}) \text{ and} \tag{4.11}$$

$$|R_{i,j}|^{(w)} = \sum_{a=1}^{|R_{i,j}|} \text{weight}(a, R_{i,j}), \tag{4.12}$$

respectively. With this notation, the first subscript ($i = 1, \ldots, m$) refers to the subset number as created by the partition on the explanatory attribute and the second subscript ($j = 1, \ldots, k$) represents the class value. Unless noted otherwise, we shall assume that all instances have a weight of 1 and write $|S|$ for $|S|^{(w)}$, $|L|$ for $|L|^{(w)}$, and $|R|$ for $|R|^{(w)}$.

The gain ratio of a split on a continuous attribute Y at threshold T of a dataset S is

$$\text{GainRatio}(Y,T;S) = \frac{\text{Entropy}(S) - (\frac{|L|}{|S|} * \text{Entropy}(L) + \frac{|R|}{|S|} * \text{Entropy}(R))}{-\frac{|L|}{|S|} * \log_2 \frac{|L|}{|S|} - \frac{|R|}{|S|} * \log_2 \frac{|R|}{|S|}} \tag{4.13}$$

where

$$\text{Entropy}(S) = -\sum_{i=1}^{k} \frac{|S_i|}{|S|} * \log_2 \frac{|S_i|}{|S|} \tag{4.14}$$

The GainRatio is computed for all candidate thresholds (i.e., all distinct values of Y in S). The threshold leading to the highest GainRatio is selected as best cut point and its gain ratio accounts for the relevance of the continuous attribute. In decision tree learning, all intermediate cumulative weights are no longer used and therefore deleted from memory. In our case, we need to keep these values since they will be required to optimize the search for transformations and to generate graphs showing the nature and the strength of the dependencies. The *relevance matrix* is the data structure we use to store the cumulative and balance sums of weights.

Each relevance matrix stores the relevance information of a response attribute for a specific dataset. For the analysis of the interaction between an explanatory and a response attribute, we need a relevance matrix for each subset associated to the values of the explanatory attribute (i.e. subsets $S_1, S_2, \ldots, S_m$) plus one for the global relevance of the response attribute (in S). Therefore, the total number of relevance matrices needed for the analysis of a pair of attributes is equal to the total number of levels of the explanatory attribute[1] plus one.

Table 4.1 shows one of the relevance matrices created during the analysis of the interaction between attributes Y_1 and X_1 from the example presented in Section 4.2.2. In this case, four relevance matrices were required; one for global relevance and one for each of the three values of X_1. The relevance matrix shown is for the subset S_3 which includes all instances s such that $\text{val}_{X_1}(s) = c$. The number of entries in a relevance matrix depends on the number of values observed for the response attribute in the given subset. Using the same example, the number of entries in the four relevance matrices are: 100 for the global dataset S, 20 for subset S_1, 42 for subset S_2, and 38 for subset S_3. Note that in this particular example, there was no duplicated value for Y_1. This is why there are as many cut points as instances. This is not generally the case.

[1]Including one for missing values, if any.

Threshold	Cumulative sums of weights $\{\|L_{3,1}\|, \|L_{3,2}\|\}$	Balance sums of weights $\{\|R_{3,1}\|, \|R_{3,2}\|\}$
8.59	{1,0}	{4,33}
13.07	{2,0}	{3,33}
* 13.19	{3,0}	{2,33}
13.62	{4,0}	{1,33}
13.71	{5,0}	{0,33}
15.84	{5,1}	{0,32}
16.00	{5,2}	{0,31}
16.40	{5,3}	{0,30}
16.41	{5,4}	{0,29}
17.00	{5,5}	{0,28}
17.14	{5,6}	{0,27}
17.35	{5,7}	{0,26}
17.77	{5,8}	{0,25}
17.88	{5,9}	{0,24}
18.72	{5,10}	{0,23}
18.76	{5,11}	{0,22}
18.86	{5,12}	{0,21}
18.87	{5,13}	{0,20}
19.66	{5,14}	{0,19}
19.79	{5,15}	{0,18}
20.89	{5,16}	{0,17}
20.95	{5,17}	{0,16}
21.00	{5,18}	{0,15}
21.26	{5,19}	{0,14}
21.33	{5,20}	{0,13}
21.45	{5,21}	{0,12}
21.98	{5,22}	{0,11}
24.67	{5,23}	{0,10}
24.79	{5,24}	{0,9}
25.66	{5,25}	{0,8}
26.35	{5,26}	{0,7}
26.97	{5,27}	{0,6}
27.44	{5,28}	{0,5}
28.52	{5,29}	{0,4}
31.90	{5,30}	{0,3}
34.29	{5,31}	{0,2}
34.83	{5,32}	{0,1}
37.28	{5,33}	{0,0}

Table 4.1: **Relevance matrix** of Y_1 in subset S_3 which includes all instances s such that $\text{val}_{X_1}(s) = c$ (see first domain presented in Section 4.2.2). The first column shows the candidate cut point thresholds. The second and third columns present the corresponding sums of weights of instances per class in the 2 subsets created by the split at the given threshold. *This point is explained in the text.

For each cut point, the relevance matrix shows the threshold value with the corresponding sums of weights of instances per class in the cumulative and balance subsets. For example, the threshold for the third cut point in Table 4.1 is 13.19. The cumulative sums of weights are $\{3, 0\}$ meaning that only 3 instances[2] satisfy the condition $\text{val}_{Y_1}(s) \leq 13.19$. Moreover, these 3 instances are all from the positive class. For the same cut point, the sums of weights for the balance subsets are $\{2, 33\}$, which indicate that the condition $\text{val}_{Y_1}(s) > 13.19$ is satisfied for 2 instances from the positive class (noted c_1) and 33 from the negative class (noted c_2).

The sum of sums of weights for the cumulative and balance subsets for a given class equals the total weight of the class in the data partition considered. Finally, the sum of all weights equals the total weight of the instances in the given data partition. Also, the total weight is the same for all cut points in a given matrix, only the distributions of the weights differ from one cut point to another.

The next section discusses how the contents of relevance matrices can be efficiently visualized. The relevance matrices also plays an important role in the feature generation phase (Chapter 5).

4.3 Relevance graph: A visual method to assess dependencies

Explanatory and response attributes can interact in many different manners. Each of these manners potentially produces a specific effect on the global relevance of the response attribute. To develop an adequate transformation, it is important to understand the characteristics of the interaction and its impact on the learning problem. This section introduces a novel visualization technique, named *relevance graph*, that we have developed to answer this need. We have implemented this technique in a Java system. This system has been used to generate all the relevance curves and relevance graphs presented in this section. Complying with the previous sections, the discussion considers only the visualization of the interactions between an explanatory and a response attribute. Section 5.3.1 discusses simple extensions to allow visualization of simultaneous effects from several explanatory attributes.

A relevance graph characterizes the interaction between the response and explanatory attributes using two types of information: the *relevance* information and the *compatibility* information. The relevance information is represented through a series of relevance curves

[2]To simplify the presentation, we assume that all instances have a weight of 1.0. Therefore, the instance count is the same as the sum of weights.

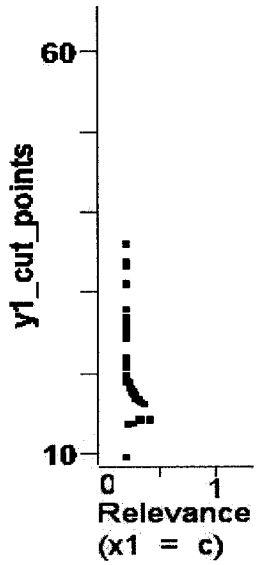


Figure 4.5: Relevance curve showing the relevance of Y_1 in the subset defined by $X_1 = c$ (see first domain presented in Section 4.2.2).

while the compatibility information is depicted using special annotations at each cut point. The next two sections describe these two types of information and the way that they appear in the relevance graph. Section 4.3.3 presents examples of relevance graphs characterizing various forms of interactions and describes their interpretations.

4.3.1 Relevance curves

A relevance curve is a simple graph showing the ability of each cut point to separate the instances in a given partition according to the class values. There is a one-to-one association between a relevance curve and a relevance matrix. The relevance matrix stores all the required information to generate the curve. Figure 4.5 shows the relevance curve associated with the relevance matrix presented in Table 4.1.

Each point on the curve stands for a cut point, or equivalently an entry in the relevance matrix. The vertical axis is for the threshold values while the horizontal axis represents the relevance of each cut point. The larger the value along the horizontal axis; the better the effect of the cut point in producing pure partitions (i.e. partitions with low entropy).

Definition 2 The *relevance* of a cut point on an attribute Y at a threshold T of a dataset S is

$$\mathrm{U}(Y,T;S) = (1.0 - \mathrm{Entropy}(S)) + \mathrm{GainRatio}(Y,T;S) \tag{4.15}$$

□

The first term represents the initial purity of the dataset S, before a cut point is applied. The second term is the gain ratio that would be obtained by splitting the partition on the given cut point.

Since the first term in Equation (4.15) is constant, the differences in relevance among cut points in a given curve are only attributed to their respective gain ratio. Therefore, cut points having a gain ratio that is equal to zero would be minimally useful. On the other hand, cut points with high gain ratios will have greater relevance values.

The distribution of the gain ratio values for the cut points is shown to be asymptotically Gaussian [Kvålseth, 1987]. Therefore, for any sufficiently large partition, the cut points located at the extremes of each relevance curve have a gain ratio close to zero. This means that the relevance of the minimal (or maximal) cut point of a relevance curve accurately represents the initial purity of the data partition. This particularity makes it easy to assess the importance of each term in Equation (4.15). For instance, to evaluate the gain ratio of a given cut point, it suffices to compare its relevance to the relevance of the minimal (or maximal) cut point. The difference in relevance between the two cut points, if any, would be due to the gain ratio of the given cut point.

In practice, the curves are rarely perfectly Gaussian which means that other cut points along a relevance curve could also have a gain ratio that is equal to zero (in particular the outer ones). Consider the extreme situation in which no cut point helps decrease the entropy of the dataset. In this case, all cut points would be at an equal distance from the vertical axis. Visually, the corresponding relevance curve would be a line parallel to the vertical axis. This particular situation happens, for example, when all instances in a given dataset belong to the same class.

The objective of the relevance graph is to allow quick visual assessment of changes in the behavior of relevance of the response attribute across groups. To make this possible, the relevance graph shows the relevance curves for all levels of the explanatory attribute simultaneously. The relevance graph also displays a relevance curve for the global dataset and one for missing values, if any. For instance, with the first domain in Section 4.2.2, the relevance graph showing the effect of X_1 on Y_1 will show four relevance curves; one for each of the three levels of X_1 and one for the global relevance of Y_1. Figure 4.6 presents this graph. Before each curve, there is a vertical *reference line*. This line stands for the vertical

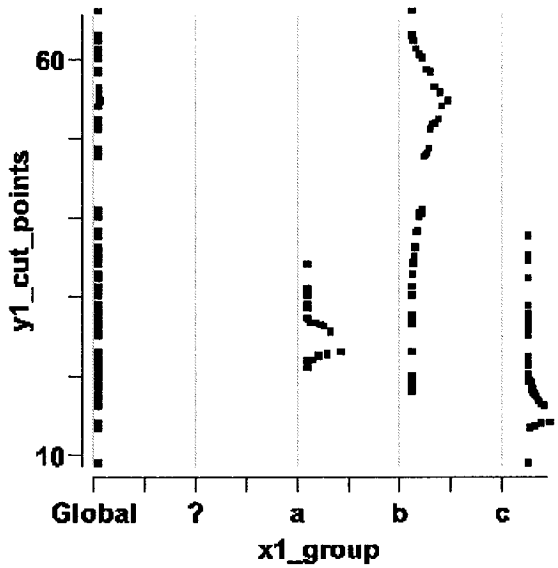


Figure 4.6: A series of relevance curves showing the relevance of Y_1 over the partitions created by splitting on the values of X_1 (see first domain presented in Section 4.2.2).

axis. Therefore, the horizontal distance between the point and the reference line represents the relevance of each cut point. In Figure 4.6, there is one additional reference line for missing values, but no relevance curve attached to it. This is because there was no missing value for attribute X_1 in the dataset.

4.3.2 Annotating compatibilities among cut points

In addition to relevance, it is important to assess the *compatibility* or *synergy* across the various subsets generated by the partition on the explanatory attribute. The notion of compatibility refers to the joint effect of cut points associated to a given threshold across subsets. The compatibility information is essential since it helps understand how a given interaction interferes with the learning task. This section explains the method we have developed to efficiently visualize the compatibility information. As we shall see, the compatibility information is incorporated by annotating each cut point on the relevance curve presented above with a specific coding that involves various shapes (or symbols) and colors. The coding we propose is supported with a theory that guarantees an unambiguous interpretation. This section details this theory. First, we will formalize the notion of *accuracy*. Second, we will introduce the definition of cut point *compatibility* and illustrate the relations

with the concept of relevance. Third, we will present two *distribution characteristics* that allow us to link the notion of compatibility to some simple visual annotations. Finally, we will present a theorem to specify the consequences of these two characteristics on cut point compatibility as well as on the interpretation of the proposed coding.

First, let us define the accuracy associated with a split of a dataset on a given cut point:

Definition 3 The *accuracy* of a cut point on an attribute Y at threshold T of a dataset S is

$$\text{Accuracy}(Y, T; S) = \frac{|L^*| + |R^*|}{|S|} \tag{4.16}$$

where

$$L^* = \{s \in L \mid \text{val}_C(s) = \text{maj}_C(L)\} \tag{4.17}$$

$$R^* = \{s \in R \mid \text{val}_C(s) = \text{maj}_C(R)\} \tag{4.18}$$

$$\text{maj}_C(S) = \arg\max_{1 \leq i \leq k} |\{s \in S \mid \text{val}_C(s) = c_i\}| \tag{4.19}$$

and where L and R are as defined in Equation (4.3). □

To put in words, L^* and R^* are simply the sets containing the instances that belong to the majority class in L and R, respectively. Accordingly, the values $|L^*|$ and $|R^*|$ correspond to the numbers of instances properly classified on the left and right side of the split, respectively. Since the value $|S|$ equals the total number of instances in the set S, Equation (4.16) says that the accuracy of a cut point is given by the the total number of instances properly classified by the split divided by the total number of instances. This complies with the standard meaning of accuracy in machine learning.

Using the above definition of accuracy, we can define the notion of cut point compatibility as follow:

Definition 4 The m cut points defined on an attribute Y at threshold T for subsets $S_1, S_2, \ldots, S_m$ are said to be *compatible* if and only if

$$\text{Accuracy}(Y, T; S) = \sum_{i=1}^{m} \frac{|S_i|}{|S|} \text{Accuracy}(Y, T; S_i) \tag{4.20}$$

and said to be *incompatible* if and only if

$$\text{Accuracy}(Y, T; S) < \sum_{i=1}^{m} \frac{|S_i|}{|S|} \text{Accuracy}(Y, T; S_i) \tag{4.21}$$

□

Example 3 Consider two cut points on an attribute Y at threshold T across subsets S_1 and S_2. Suppose that the relevance matrix entries for these two cut points are $< T, \{5,1\}, \{4,14\} >$ and $< T, \{10,1\}, \{1,6\} >$, respectively. If S_1, S_2 is a partition of S, then the corresponding relevance matrix entry at the global level is $< T, \{15,2\}, \{5,20\} >$. Using the notation introduced in Section 4.3.1, we have

$$|S_1| = 6 + 18 = 24 \qquad |S_2| = 11 + 7 = 18 \qquad |S| = 17 + 25 = 42$$

and the accuracies are

$$\begin{aligned} \text{Accuracy}(Y, T; S_1) &= (5 + 14)/24 \\ \text{Accuracy}(Y, T; S_2) &= (10 + 6)/18 \\ \text{Accuracy}(Y, T; S) &= (15 + 20)/42 \end{aligned}$$

Substituting in the left hand side (LHS) and right hand side (RHS) of Equation 4.20, we obtain

$$\begin{aligned} LHS &= RHS \\ \frac{15+20}{42} &= \left(\frac{24}{42}\right)\left(\frac{5+14}{24}\right) + \left(\frac{18}{42}\right)\left(\frac{10+6}{18}\right) \\ \frac{15+20}{42} &= \frac{5+14+10+6}{42} \approx 0.83 \end{aligned}$$

Therefore, the cut points are *compatible*. Also note that the global relevance of the split $\text{U}(Y, T; S) \approx 0.368$. □

Example 4 As another example, suppose we replace the second cut point in Example 3 by $< T, \{1,10\}, \{1,6\} >$ instead of $< T, \{10,1\}, \{1,6\} >$. The corresponding relevance matrix entry at the global level is $< T, \{6,11\}, \{5,20\} >$. This change does not affect the values for $\text{Accuracy}(Y, T; S_1)$ and $\text{Accuracy}(Y, T; S_2)$. However, the global accuracy becomes

$$\text{Accuracy}(Y, T; S) = (11 + 20)/42 \approx 0.74$$

Since

$$\left(\frac{11+20}{42} \approx 0.74\right) < \left(\frac{5+14+10+6}{42} \approx 0.83\right)$$

the cut points are *incompatible* (Equation (4.21)). The resulting global relevance of the split

is now $U(Y,T;S) \approx 0.192$, which is significantly less than the initial one (i.e., 0.368). □

As illustrated by the above examples, incompatibility could have a great effect on the global relevance. It is therefore important to identify incompatible cut points during the analysis of interactions. However, direct application of the above procedure for all pairs of cut points is not practical since: (i) the number of such pairs is exponential in the number of partitions and (ii) it would be very difficult to efficiently visualize the results for all pairwise comparisons on the same graph (there would be too much information to display). We avoid these difficulties by linking the notion of compatibility to the characteristics of the distribution of the instances around each cut point. We define two distribution characteristics. These two characteristics are then depicted on the relevance graph using simple symbols, thus making it easy for the analyst to quickly find the incompatible cut points.

Definition 5 The *distribution characteristics 1 and 2*, of a cut point on an attribute Y at threshold T of a dataset S are

$$\lambda_1(Y,T;S) = \text{maj}_C(L) = \arg\max_{1\leq i\leq k} |\{s \in L \mid \text{val}_C(s) = c_i\}| \tag{4.22}$$

$$\lambda_2(Y,T;S) = \text{maj}_C(R) = \arg\max_{1\leq i\leq k} |\{s \in R \mid \text{val}_C(s) = c_i\}| \tag{4.23}$$

Remark 1 If more than one class value satisfies $\text{maj}_C(L)$, then we say that $\lambda_1(Y,T;S)$ is undetermined. Similarly, $\lambda_2(Y,T;S)$ is said to be undetermined if more than one class value satisfies $\text{maj}_C(R)$. Unless noted otherwise, we shall assume that both $\lambda_1(Y,T;S)$ and $\lambda_2(Y,T;S)$ are determined. □

In words, the characteristic $\lambda_1(Y,T;S)$ refers to the majority class among the instances falling on the left hand side of the split. Similarly, $\lambda_2(Y,T;S)$ refers to the majority class among the instances falling on the right hand side of the split. When there is no risk of confusion, we write λ_1 for $\lambda_1(Y,T;S)$ and λ_2 for $\lambda_2(Y,T;S)$. As indicated by the remark following the definition, a special case happens when the majority class is not unique.

To compute the distribution characteristics, we use the cumulative weights available in the relevance matrix. Precisely, for each cut point, we identify the majority class in subsets L and R and set λ_1 and λ_2 accordingly.

The number of possible values for the two characteristics combined equals k^2, where k is the the number of class values. For instance, in a problem with two classes values $\{+,-\}$, there are four possibilities as illustrated in Table 4.2. Considering the compatibility of a group of m cut points, there are $\sum_{i=0}^{km} = (km+1)km/2$ unique possible assignments

Characteristic 1 ($\lambda_1 = \text{maj}_C(L)$)	Characteristic 2 ($\lambda_2 = \text{maj}_C(L)$)
+	+
+	−
−	+
−	−

Table 4.2: Possible values for distribution characteristics 1 and 2 for a two-class problem.

for their joint distribution characteristics. For example, in a two class problem with values $\{+, -\}$ and partition size $m = 2$, there are 10 distinct assignments for their joint distribution characteristic values:

$$\{<+,+>,<+,+>\}$$
$$\{<+,+>,<+,->\}$$
$$\{<+,+>,<-,+>\}$$
$$\{<+,+>,<-,->\}$$
$$\{<+,->,<+,->\}$$
$$\{<+,->,<-,+>\}$$
$$\{<+,->,<-,->\}$$
$$\{<-,+>,<-,+>\}$$
$$\{<-,+>,<-,->\}$$
$$\{<-,->,<-,->\}$$

How can we quickly identify the compatible cut points among these possibilities? The following theorem answers this question.

Theorem 1 *The m cut points defined on an attribute Y at threshold T for subsets $S_1, S_2, \ldots, S_m$, a partition of S, are compatible if and only if they have identical values for distribution characteristics 1 and 2, that is:*

$$\lambda_1(Y,T;S_i) = \lambda_1(Y,T;S_j) \tag{4.24}$$

$$\lambda_2(Y,T;S_i) = \lambda_2(Y,T;S_j) \tag{4.25}$$

for all $i,j \in \{1,2,\ldots,m\}$ and $i \neq j$.

PROOF We proceed by negation. Suppose the above proposition is false, then there must exist a subset S_p such that

$$\lambda_1(Y,T;S_p) \neq \lambda_1(Y,T;S_i) \quad \text{or} \quad \lambda_2(Y,T;S_p) \neq \lambda_2(Y,T;S_i)$$

for $i \in \{1,2,\ldots,p-1,p+1,p+2,\ldots,m\}$. There are three ways to satisfy the above condition. First, $\lambda_1(Y,T;S_p)$ is distinct but $\lambda_2(Y,T;S_p)$ is identical to the other λ_2's. Second, $\lambda_2(Y,T;S_p)$ is distinct but $\lambda_1(Y,T;S_p)$ is identical to the other λ_1's. Third, both $\lambda_1(Y,T;S_p)$ and $\lambda_2(Y,T;S_p)$ are distinct. We start by proving the non-existence of a distinct λ_1 and then apply the same approach for λ_2. Finally, we address the case of distinct λ_1 and λ_2. To simplify the notation, we shall work on a simpler but equivalent partition S'_1, S'_2 of S such that $S'_1 = S_p$ and $S'_2 = S_1 \cup S_2 \cup \cdots \cup S_{(p-1)} \cup S_{(p+1)} \cup \cdots \cup S_m$.

A distinct λ_1: Let $\lambda_1(Y,T;S'_1) = p$, $\lambda_1(Y,T;S'_2) = q$, and $\lambda_2(Y,T;S'_1) = \lambda_2(Y,T;S'_2) = u$. Using this assignment and the notation introduced in Equation (4.5) for cumulative and balance sets, the right side of Equation (4.20) is

$$\begin{aligned}
\sum_{i=1}^{m} \frac{|S_i|}{|S|} \text{Accuracy}(Y,T;S_i) &= \frac{|S'_1|}{|S|} \text{Accuracy}(Y,T;S'_1) + \frac{|S'_2|}{|S|} \text{Accuracy}(Y,T;S'_2) \\
&= \frac{|S'_1|}{|S|}\left(\frac{|L_{1,p}| + |R_{1,u}|}{|S'_1|}\right) + \frac{|S'_2|}{|S|}\left(\frac{|L_{2,q}| + |R_{2,u}|}{|S'_2|}\right) \\
&= \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,q}| + |R_{2,u}|}{|S|}
\end{aligned}$$

For the left side of Equation (4.20), we have two possibilities:

Case 1: $\lambda_1(Y,T;S) = p$, then

$$\text{Accuracy}(Y,T;S) = \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,p}| + |R_{2,u}|}{|S|}$$

To have *compatibility*, we need equality of both sides of Equation (4.20), hence

$$\frac{|L_{1,p}| + |R_{1,u}| + |L_{2,p}| + |R_{2,u}|}{|S|} = \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,q}| + |R_{2,u}|}{|S|}$$

$$|L_{2,p}| = |L_{2,q}|$$

Which is not possible since, by Definition 4, $|L_{2,p}| < |L_{2,q}|$.

Case 2: $\lambda_1(Y,T;S) = q$, then

$$\text{Accuracy}(Y,T;S) = \frac{|L_{1,q}| + |R_{1,u}| + |L_{2,q}| + |R_{2,u}|}{|S|}$$

To have *compatibility*, we need equality of both sides of Equation (4.20), hence

$$\frac{|L_{1,q}| + |R_{1,u}| + |L_{2,q}| + |R_{2,u}|}{|S|} = \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,q}| + |R_{2,u}|}{|S|}$$

This is also impossible since, by Definition 4, $|L_{1,q}| < |L_{1,p}|$.

A distinct λ_2: By applying the above approach, we also find that it is impossible to have a distinct λ_2 and still achieve compatibility.

Distinct λ_1 and λ_2: Let $\lambda_1(Y,T;S'_1) = p$, $\lambda_1(Y,T;S'_2) = q$, $\lambda_2(Y,T;S'_1) = u$, and $\lambda_2(Y,T;S'_2) = v$. Using this assignment and the notation introduced in Equation (4.5) for cumulative and balance sets, the right side of Equation (4.20) is

$$\begin{aligned}
\sum_{i=1}^{m} \frac{|S_i|}{|S|} \text{Accuracy}(Y,T;S_i) &= \frac{|S'_1|}{|S|} \text{Accuracy}(Y,T;S'_1) + \frac{|S'_2|}{|S|} \text{Accuracy}(Y,T;S'_2) \\
&= \frac{|S'_1|}{|S|}\left(\frac{|L_{1,p}| + |R_{1,u}|}{|S'_1|}\right) + \frac{|S'_2|}{|S|}\left(\frac{|L_{2,q}| + |R_{2,v}|}{|S'_2|}\right) \\
&= \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,q}| + |R_{2,v}|}{|S|}
\end{aligned}$$

For the left side of Equation (4.20), we have four possibilities:

Case 1: $\lambda_1(Y,T;S) = p$ **and** $\lambda_2(Y,T;S) = u$, then

$$\text{Accuracy}(Y,T;S) = \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,p}| + |R_{2,u}|}{|S|}$$

To have *compatibility*, we need equality of both sides of Equation (4.20), hence

$$\frac{|L_{1,p}| + |R_{1,u}| + |L_{2,p}| + |R_{2,u}|}{|S|} = \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,q}| + |R_{2,v}|}{|S|}$$

Which is not possible since, by Definition 4, we have $|L_{2,p}| < |L_{2,q}|$ and $|R_{2,u}| < |R_{2,v}|$.

Case 2: $\lambda_1(Y,T;S) = p$ **and** $\lambda_2(Y,T;S) = v$, then

$$\text{Accuracy}(Y,T;S) = \frac{|L_{1,p}| + |R_{1,v}| + |L_{2,p}| + |R_{2,v}|}{|S|}$$

To have *compatibility*, we need equality of both sides of Equation (4.20), hence

$$\frac{|L_{1,p}| + |R_{1,v}| + |L_{2,p}| + |R_{2,v}|}{|S|} = \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,q}| + |R_{2,v}|}{|S|}$$

Which is not possible since, by Definition 4, we have $|L_{2,p}| < |L_{2,q}|$ and $|R_{1,v}| < |R_{1,u}|$.

Case 3: $\lambda_1(Y,T;S) = q$ **and** $\lambda_2(Y,T;S) = u$, then

$$\text{Accuracy}(Y,T;S) = \frac{|L_{1,q}| + |R_{1,u}| + |L_{2,q}| + |R_{2,u}|}{|S|}$$

To have *compatibility*, we need equality of both sides of Equation (4.20), hence

$$\frac{|L_{1,q}| + |R_{1,u}| + |L_{2,q}| + |R_{2,u}|}{|S|} = \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,q}| + |R_{2,v}|}{|S|}$$

Which is not possible since, by Definition 4, we have $|L_{1,q}| < |L_{1,p}|$ and $|R_{2,u}| < |R_{2,v}|$.

Case 4: $\lambda_1(Y,T;S) = q$ **and** $\lambda_2(Y,T;S) = v$, then

$$\text{Accuracy}(Y,T;S) = \frac{|L_{1,q}| + |R_{1,v}| + |L_{2,q}| + |R_{2,v}|}{|S|}$$

To have *compatibility*, we need equality of both sides of Equation (4.20), hence

$$\frac{|L_{1,q}| + |R_{1,v}| + |L_{2,q}| + |R_{2,v}|}{|S|} = \frac{|L_{1,p}| + |R_{1,u}| + |L_{2,q}| + |R_{2,v}|}{|S|}$$

Which is not possible since, by Definition 4, we have $|L_{1,q}| < |L_{1,p}|$ and $|R_{1,v}| < |R_{1,u}|$.

Thus we have shown that cut points with identical distribution characteristics are compatible. ■

Theorem 1 implies that the two proposed class distribution characteristics are sufficient to determine the compatibility between any group of cut points. To visually provide the required information about these two distribution characteristics, we annotate each cut point in the relevance graph with two visual signs. First, the *color* symbolizes the first characteristic; a distinct color is used for each class value. A specific color is reserved to represent undetermined λ_1. Second, the *shape* (e.g., plus, minus, cross, star, circle, box) represents the possible values of the second characteristic. Again, there is one shape for each class value plus a special shape (i.e., a triangle) for the undetermined situation. Using these notations, compatible cut points have both the same color and the same shape. Other possibilities designate incompatible cut points.

Applying these notations to the relevance curves of Figure 4.6, we obtain the relevance graph shown in Figure 4.7. Textual annotations have been added to graph in this figure to summarize the various components we have discussed. We now turn to the interpretation of relevance graphs by explaining the visual patterns that we are looking for to identify important interactions.

4.3.3 Identification of dependencies using relevance graphs

As explained in the previous sections, relevance graphs depict two kinds of information: relevance information and compatibility information. To identify the dependency relationships that are likely to hurt the learning process, the researcher needs to consider these two kinds of information concurrently. Moreover, it is required to contrast the information from local and global subsets. The analysis is performed by examining possible thresholds to pinpoint situations with non-optimal global relevance that is due to interactions across levels of the explanatory attribute. Various types of interactions can lead to this problem. We first present four types of interactions, then we investigate mixed interactions.

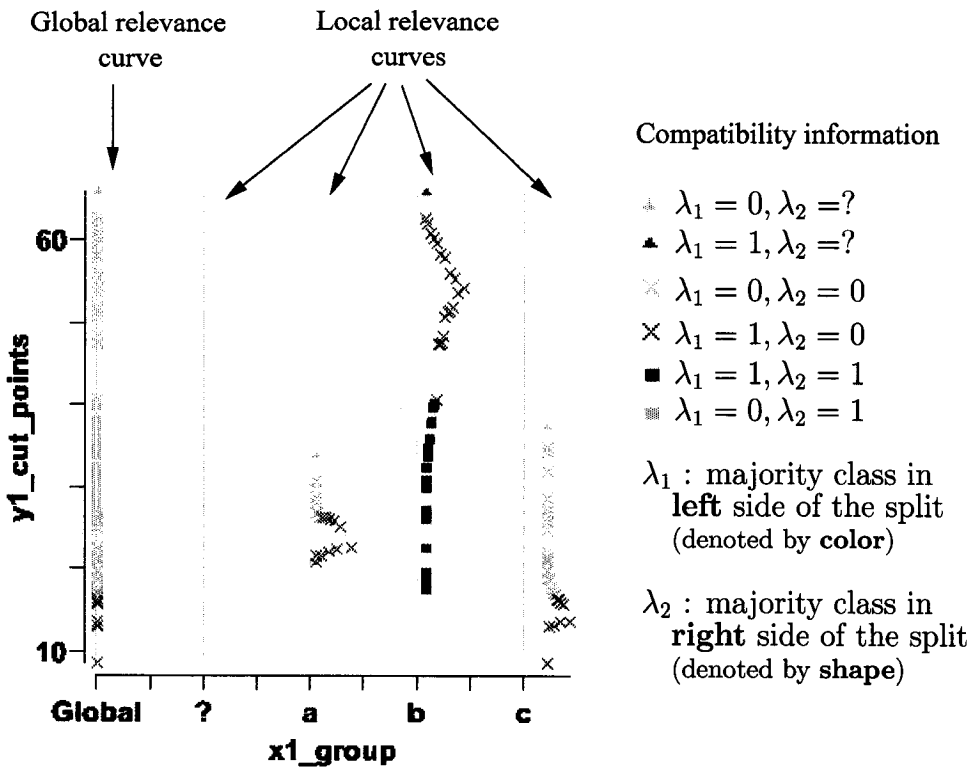


Figure 4.7: Relevance graph to analyze the effect of X_1 on the relevance of Y_1 (see first domain presented in Section 4.2.2).

Highly relevant but incompatible cut points

The first and most problematic situation happens at thresholds for which the corresponding cut points in local partitions are highly relevant but interact in an incompatible manner. This produces a reduced global relevance with respect to the subsets of the partition. The severity of the reduction in the global relevance is directly proportional to the individual relevance of the incompatible cut points.

Figure 4.8 provides an example of this kind of interaction. In this example, corresponding cut points across subsets have identical relevances but they are incompatible. The incompatibility is noticeable from the fact that all cut points have either a different shape or a different color (see Theorem 1).

The global relevance curve clearly reports the negative consequence of the interaction; all points on this curve are completely useless. In other words, from a global univariate

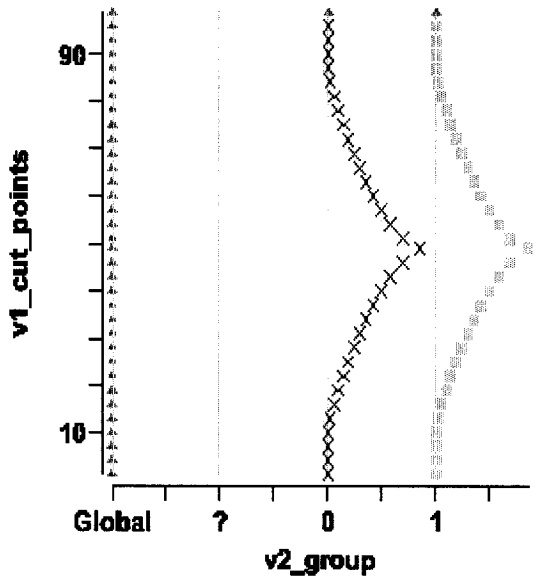


Figure 4.8: Example of highly relevant but incompatible cut points. All the local relevances are cancelled by the interactions.

perspective, the Y attribute appears irrelevant to the learning problem. However, when the same attribute is jointly considered with attribute X, then its usefulness becomes clear. All learning algorithms that consider attributes individually may miss this relationship. The risk of not detecting such a useful attribute is particularly important in applications that are characterized by a high number of low-level attributes. The relevance graphs could be used in these applications to guide the selection of appropriate learning algorithms.

Large difference between cut point relevances

A second situation happens when the cut points are compatible but with large differences between their relevance. In this case, the resulting global relevance is less than the maximal local relevance value. The importance of the reduction depends on the differences between relevances and on the total weight of subset(s) with a low relevance. As shown in Figure 4.9, this phenomenon may have a significant impact on the global relevance observed. When performing a global univariate analysis of the attribute, this may lead to the wrong conclusion that an attribute is of little value.

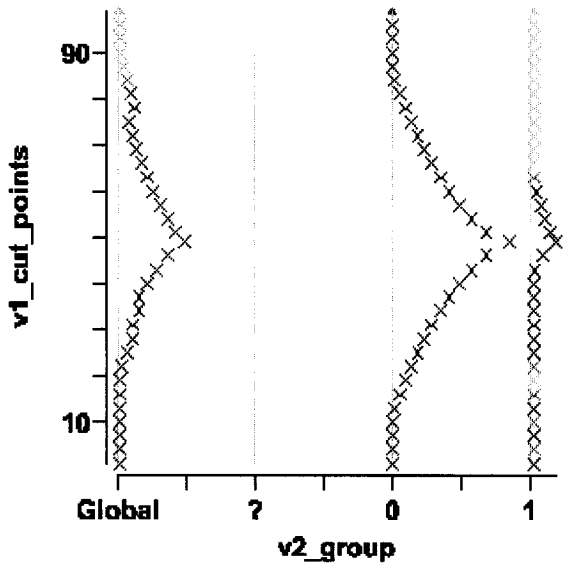


Figure 4.9: Example of attribute interactions involving cut points with different relevances.

Misaligned peak cut points

The first domain presented in Section 4.2.2 provides a perfect example of misaligned cut points. Looking at the corresponding relevance graph shown in Figure 4.7, we observed very similar patterns in the three subsets of the partitions. However, the peaks are not well aligned. This produces an important reduction of the global relevance as shown by the global relevance curve in the same figure.

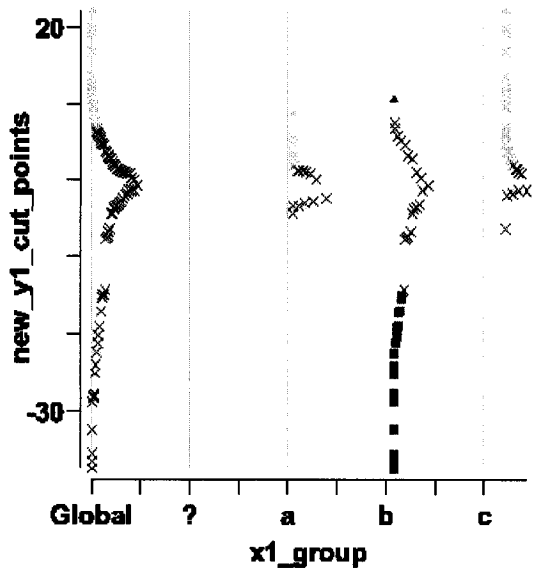


Figure 4.10: Relevance graph showing the effect of the transformation applied to Y_1 (see first domain presented in Section 4.2.2).

The transformation we applied to this domain in Section 4.2.2 was meant to correct this alignment problem. The relevance graph in Figure 4.10 shows how this transformation improved the global relevance of the attribute. The transformation perfectly aligned the local best cut points which produces a global cut point of very high relevance. With the new feature, named Y_1', there is no need to consider the value of X_1; All the information is directly available in Y_1'. For learning algorithms that analyze the attributes individually such as C4.5, the use of Y_1' will ensure that they find the proper information earlier in the process. This should positively influence the final accuracy and should help reduce the complexity of the model.

Favorable interactions

In Section 4.2.2, we also introduced a second domain in which the relationship between attributes Y_2 and X_2 was identical to the one between Y_1 and X_1. However, the class values (or labels) were different and so was the usefulness of the proposed transformation. Comparing the relevance graphs for these domains (Figure 4.11 (a) and (b) for domains 1 and 2, respectively), we now see how important is the difference between the interactions in the two domains.

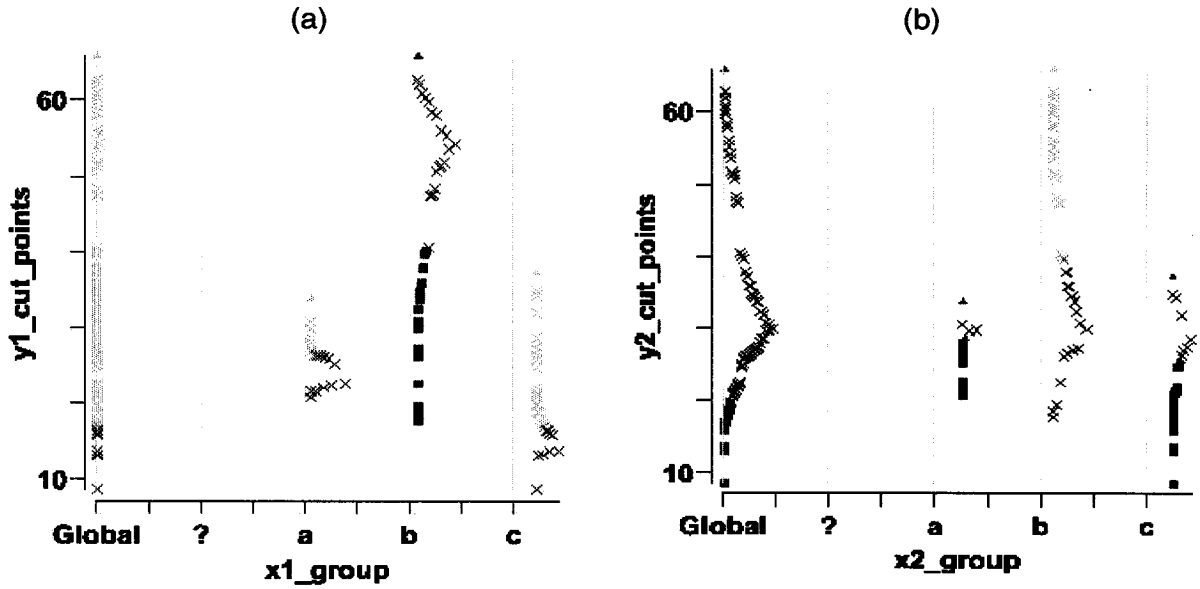


Figure 4.11: Relevance graphs for the two sample domains introduced in Section 4.2.2. Figure (a) shows that the interactions between Y_1 and X_1 is reducing the global relevance while Figure (b) shows that the interactions between Y_2 and X_2 produce a higher global relevance.

For the first domain, the relationship produced a misalignment of the most relevant local cut points. In the second domain, the same relationship did not increase the learning difficulty since the best local cut points are perfectly aligned and the corresponding global cut point is highly relevant. The relevance graph shows that any transformation on this domain would make the learning more difficult by decreasing the global relevance. This explains why the proposed transformation was not suitable for this domain. As discussed in Section 4.2.2, it was not possible to assess the difference between the two domains using a classical statistical method to analyze dependency. The relevance graphs resolve this problem.

In conclusion, the interaction observed in the domain is called a positive interaction since it simplifies the learning task. It is essential to be able to recognize these situations so that the transformation effort focuses on the problematic attribute interactions.

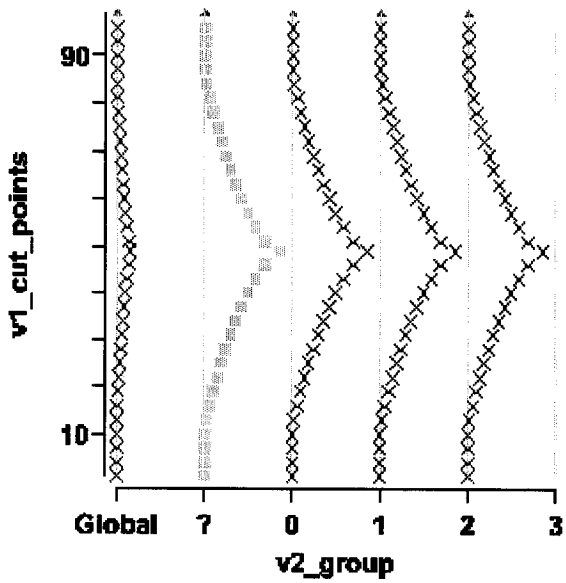


Figure 4.12: Example of attribute interactions involving cut points with different relevances.

Mixture of interactions

In practice, it is common to obtain a relevance graph showing a variety of interactions for a given threshold. For instance, with an exploratory attribute having four levels, it is possible to find at a given threshold three cut points that are compatible and one that is not. In this situation, the negative effect of the unsuitable interaction is generally so important that the phenomenon is easily detected by examining the global level. Figure 4.12 illustrates this situation; three of the four groups are perfectly cooperating but the incompatible one reduces the global relevance to almost nothing. Again, detecting such interactions is important to optimize the learning process.

4.4 Interactive relevance graphs

One of the objectives of the analysis of attribute interactions is to help find out what type of transformations are required to enhance the performance in the learning phase. The examples of last section showed that the relevance graphs provide this kind of information. However, we believe that, in the context of explanatory analysis, allowing the researcher to interactively explore some transformations could enrich the experience. We have incorporated this idea into the system developed to generate the relevance graphs.

The system provides an interface with the relevance graph and the tools required for transforming the data, as shown in Figure 4.13. There are four sections in this interface. The section in the upper left corner shows the initial threshold and relevance of the best cut point and the current best threshold with its relevance. The upper right corner shows the global relevance curve. On this graph, the vertical axis represents the relevance while the horizontal axis refers to the various thresholds of the response variable. The lower right corner presents the relevance graph as explained in Section 4.3. The tools for transforming the data are on the lower left corner. There is one slide bar for each local partition. If there is no missing value, then the first slide bar is deactivated. Moving the slide bar towards the top is equivalent to adding a constant to each cut point in the corresponding partition. Inversely, moving towards the bottom adds a negative constant. The value of this constant is proportional to the movement applied using the slide bar. The slide bars make it easy to reproduce the transformation applied to the first domain in Section 4.2.2.

Initially, the default and current cut points are identical since no transformation has been applied yet. Figures 4.13 presents this initial situation. As the researcher moves the slide bars, all other sections of the interface are dynamically updated. For instance, after applying a transformation using the second slide bar, the interface becomes as shown in Figure 4.14. We now see that the current cut point has a higher relevance than the initial one. The global relevance curve is also different. Finally, the transformation applied in Section 4.2.2 leads to the interface of Figure 4.15, which shows a much-improved global relevance by comparison to the initial situation.

Using the menus, the system can also display the current transformation model and detailed numerical values about each partition. Moreover, the menu `Transform` automates the search for the best transformation using a variety of search algorithms we have developed for this task. During the search, the information is continuously updated so that the researcher can visualize the effects of the transformation analyzed by the algorithms. These algorithms shall be presented in the next Chapter.

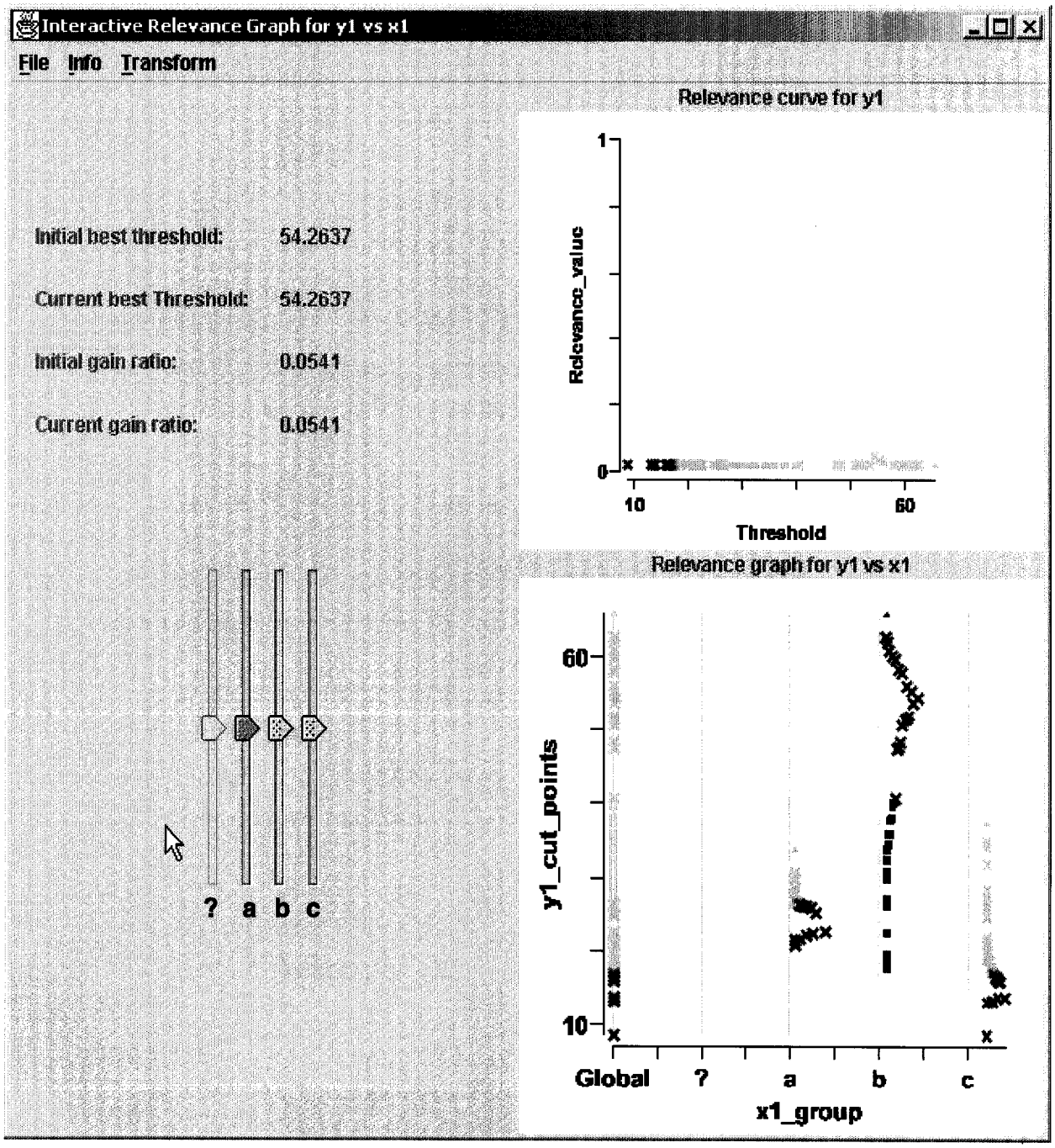


Figure 4.13: Initial interface for the interactive relevance graph program. At this point, the default and current information are identical since no transformation has been applied yet.

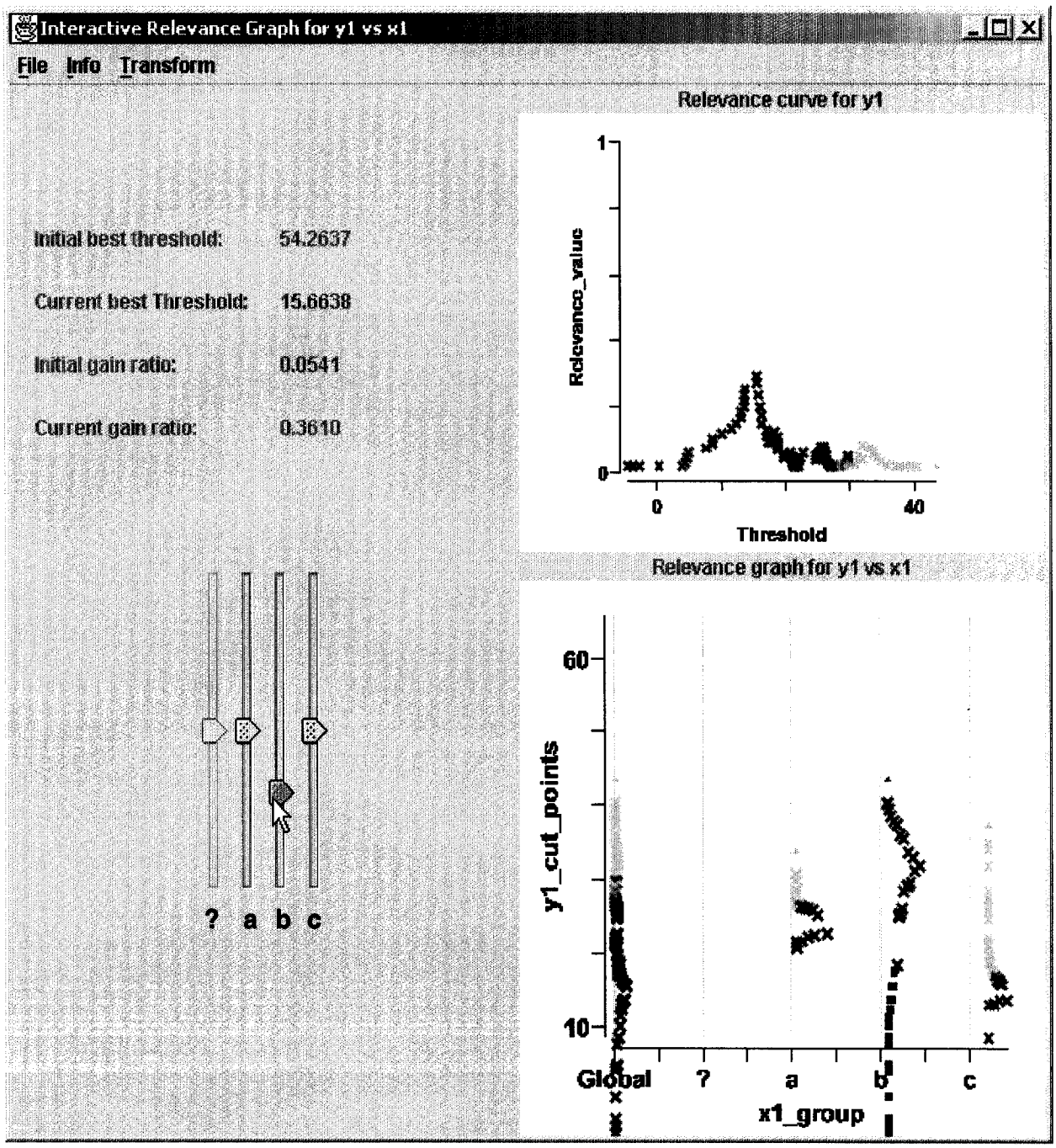


Figure 4.14: The interactive relevance graph after a transformation of the second partition.

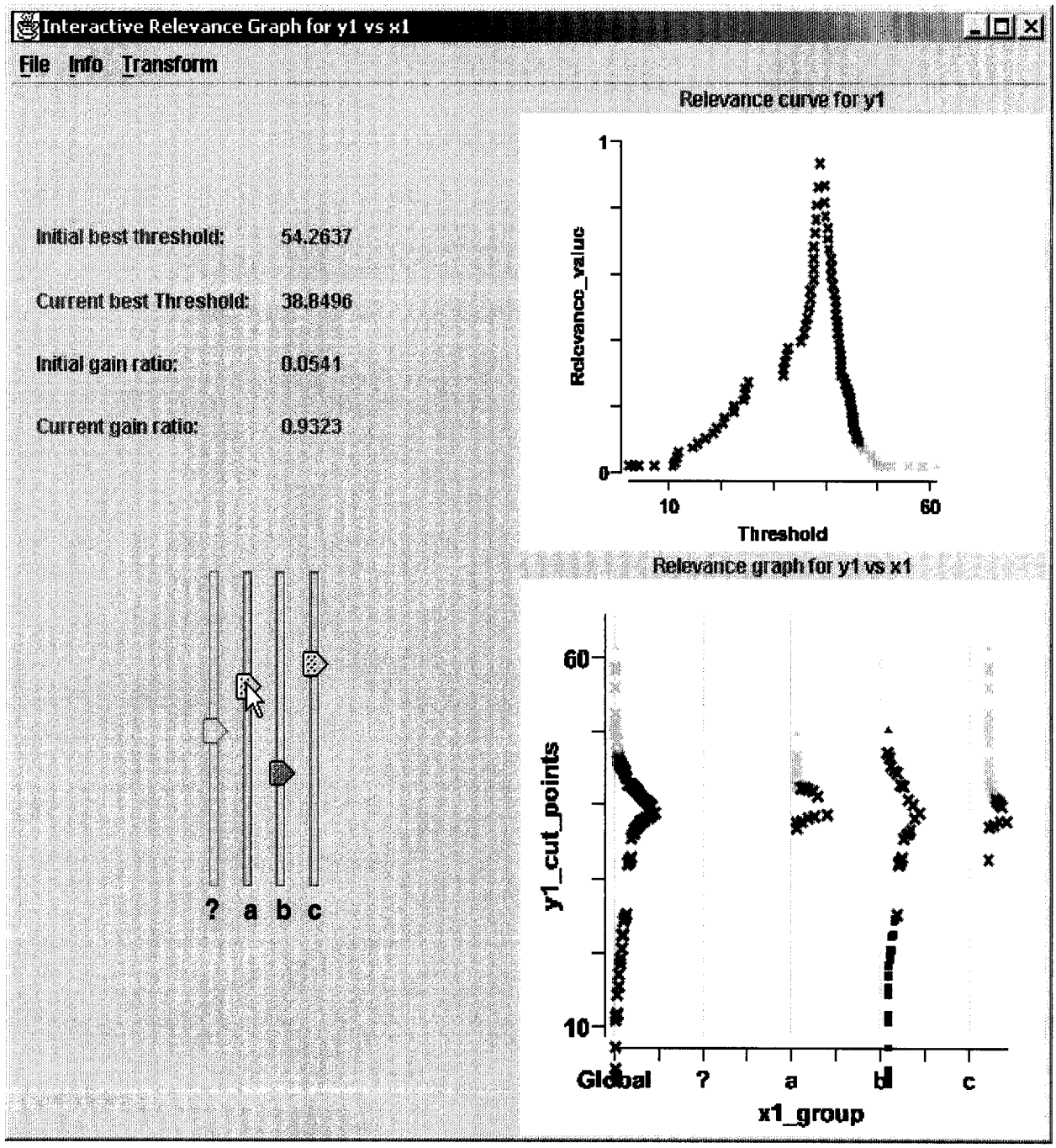


Figure 4.15: The interactive relevance graph after applying transformation described in Section 4.2.2.

4.5 Summary

Not all attribute interactions have the same effects on the results of the learning systems. To efficiently address the problem of decrease in performance, we need to understand how the attribute interactions change the learning difficulties. Classical unsupervised techniques to analyse dependencies are not appropriate for this task since they do not take into account the class information. The technique introduced in this chapter overcomes this difficulty by linking the analysis of attribute interactions to the relevance information.

After describing the computation and data structures required for the analysis of relevance, we presented a novel visualization technique named *relevance graphs*. The relevance graphs allow interactive exploration of the nature and strength of the attribute interactions in a dataset. We formally defined the properties of the relevance graphs. These properties allow a precise interpretation of the graphs. Using our implementation of the technology, we discussed the relevance graphs for a number of datasets. These examples illustrated several forms of interactions with their impact on the learning problem at hand. Finally, we proposed an extension of the relevance graphs for the interactive discovery of useful transformations. The next chapter concentrates on the second phase of the GLOREF approach which performs automatic generation of features.

Chapter 5

Generation of globally relevant features

5.1 Introduction

The previous chapter presented the first phase of the GLOREF approach which is the analysis of the effects of attribute interactions on the relevance. This chapter presents the second phase of the GLOREF approach. This phase generates numerical features that have a higher global relevance than the initial ones. The increase in global relevance is achieved by transforming the initial numerical attributes such that the negative effects of the attribute interactions are cancelled.

In Section 4.2.2, we used the following univariate transformation to cancel the negative effects of X_1 on the relevance of Y_1:

$$\Gamma(X_1, Y_1) = \begin{cases} \text{val}_{Y_1}(s) - 25.0 & \text{if } \text{val}_{X_1}(s) = a \\ \text{val}_{Y_1}(s) - 55.0 & \text{if } \text{val}_{X_1}(s) = b \\ \text{val}_{Y_1}(s) - 15.0 & \text{if } \text{val}_{X_1}(s) = c \end{cases}$$

The feature resulting from this transformation has a gain ratio that is significantly higher than the gain ratio of the initial attribute Y_1 (i.e., ≈ 0.93 and ≈ 0.05, respectively). Sections 4.2.2 and 4.3.3 further discussed the usefulness of this transformation. The relevance graphs introduced in the previous chapter can be used to discover the need for such transformations. We also discussed how the interactive relevance graph system could be used to find the set of coefficients in the above expression. In this chapter, we introduce a feature generation technique that automatically builds such transformations. The proposed approach also addresses multivariate transformations in which the joint effects of several explanatory attributes are considered.

Figure 4.1 (Section 4.1) sketched the main phases of the GLOREF approach which are the *Analysis of Relevance* and the *Feature Generation*. Figure 5.1 further details these

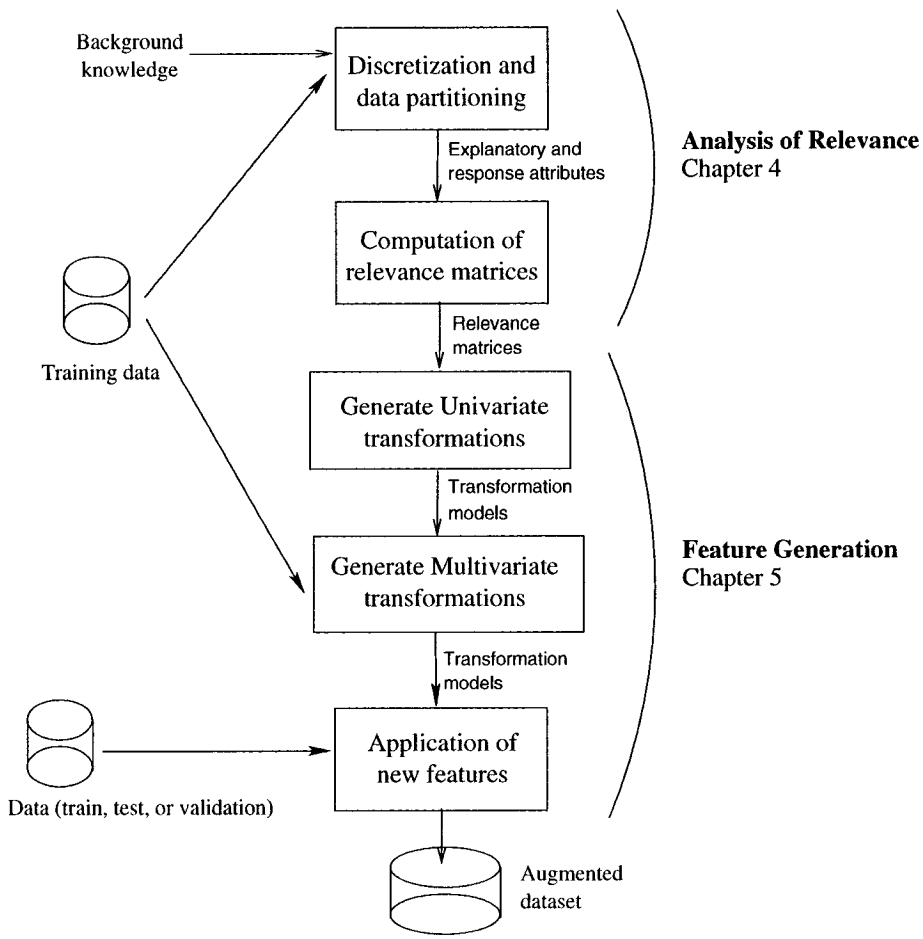


Figure 5.1: Overview of the main steps involved in the GLOREF approach.

phases by presenting the five tasks shaping the GLOREF approach. The first two tasks are part of the analysis of relevance and have been explained in Chapter 4 (Sections 4.2.3 and 4.2.4). This chapter concentrates on the last three tasks. There are five additional sections in this chapter. The first one formalizes the univariate and multivariate transformation problems. The following three sections detail our solution for the three tasks in the feature generation phase. Precisely, Section 5.2 presents the creation of univariate transformations, Section 5.3 addresses the more complicated case of multivariate transformations, and Section 5.4 discusses the application of the transformation models. Finally, Section 5.5 summarizes this chapter.

5.1.1 The problem statement

Building on the notation of Section 4.2.1, the objective is to automatically derive new numerical features $Z_1, Z_2, \ldots, Z_r$ that are highly relevant according to the relevance function U as defined by Equation (4.15). The result is an augmented representation space

$$\{X_1, X_2, \ldots, X_p, Y_1, Y_2, \ldots, Y_q, Z_1, Z_2, \ldots, Z_r, C\}$$

that contains all the initial attributes plus the new features. For simplicity, we suppose that an attribute X_i could be either one of the initial nominal attributes or a discretized attribute automatically generated by the discretization procedure described in Section 4.2.3. The attributes Y_i are the initial numerical attributes and C is the class attribute. As in Chapter 4, the attributes X and Y are named the *explanatory* and *response* attributes, respectively. The approach automatically determines the number r of new features $(Z_1, Z_2, \ldots, Z_r)$. Each feature is derived by applying a transformation $\Gamma(\mathbf{X}, Y; \Omega, s)$ where $\mathbf{X} \subseteq \{X_1, X_2, \ldots, X_p\}$, $Y \in \{Y_1, Y_2, \ldots, Y_q\}$, Ω is the set of parameters, and $s \in S$.

We distinguish two forms of transformations according to the cardinality of $\mathbf{X}$. If $|\mathbf{X}| = 1$, then we talk about a univariate transformation denoted $\Gamma(X, Y; \Omega, s)$. A univariate transformation involves only one explanatory attribute. If $|\mathbf{X}| > 1$, then $\Gamma(\mathbf{X}, Y; \Omega, s)$ is a multivariate transformation involving several explanatory attributes.

Univariate transformation

Supposing that $\mathrm{Dom}(X) = \{X(1), X(2), \ldots, X(m)\}$, then the values for a new feature Z derived from a *univariate* transformation are

$$\mathrm{val}_Z(s) = \Gamma(X, Y; \Omega, s) = \begin{cases} \mathrm{val}_Y(s) + \omega_1 & \text{if } \mathrm{val}_X(s) = X(1) \\ \mathrm{val}_Y(s) + \omega_2 & \text{if } \mathrm{val}_X(s) = X(2) \\ \vdots & \\ \mathrm{val}_Y(s) + \omega_m & \text{if } \mathrm{val}_X(s) = X(m) \end{cases} \tag{5.1}$$

where $\mathrm{Dom}(Z) = \mathbb{R}$, $\omega_i \in \mathbb{R}$ for $i \in \{1, 2, \ldots, m\}$, and $s \in S$. To simplify, we use the short hands Z for $\mathrm{val}_Z(s)$, X for $\mathrm{val}_X(s)$, Y for $\mathrm{val}_Y(s)$, and $\Gamma(X, Y; \Omega)$ for $\Gamma(X, Y; \Omega, s)$. Using

these abbreviations, the above equation becomes

$$Z = \Gamma(X, Y; \Omega) = \begin{cases} Y + \omega_1 & \text{if } X = X(1) \\ Y + \omega_2 & \text{if } X = X(2) \\ \vdots & \\ Y + \omega_m & \text{if } X = X(m) \end{cases} \tag{5.2}$$

There is an infinite number of sets of parameters $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$ that can be used to generate Z. The objective is to find the set of parameters Ω^* such that

$$\Omega^* = \arg\max_{\Omega \in \mathbb{R}^m} \mathrm{U}(Z, T^*; S) \tag{5.3}$$

where T^* is the optimal threshold for a split of S using feature Z.

Multivariate transformation

The general definition of a *multivariate* transformation involving l explanatory attributes $X_1, \ldots, X_l$, with $|\mathrm{Dom}(X_i)| = m_i$, for $i = \{1, 2, \ldots, l\}$, is

$$Z = \Gamma(\mathbf{X}, Y; \Omega) = \begin{cases} Y + \omega_1 & \text{if } X_i = X_i(1) \text{ for } i = 1, 2, \ldots, l \\ Y + \omega_2 & \text{if } X_1 = X_1(2), X_i = X_i(1) \text{ for } i = 2, 3, \ldots, l \\ \vdots & \\ Y + \omega_{m_1} & \text{if } X_1 = X_1(m_1), X_i = X_i(1) \text{ for } i = 2, 3, \ldots, l \\ Y + \omega_{m_1+1} & \text{if } X_2 = X_2(2), X_i(s) = X_i(1) \text{ for } i = 1, 3, 4, \ldots, l \\ Y + \omega_{m_1+2} & \text{if } X_1 = X_1(2), X_2 = X_2(2), X_i = X_i(1) \text{ for } i = 3, 4, \ldots, l \\ \vdots & \\ Y + \omega_M & \text{if } X_i = X_i(m_i) \text{ for } i = 1, 2, \ldots, l \end{cases} \tag{5.4}$$

where $\mathrm{Dom}(Z) = \mathbb{R}$, $M = m_1 * m_2 * \cdots * m_l$, and $\omega_j \in \mathbb{R}$ for $j \in \{1, 2, \ldots, M\}$. The conditions in the above expression define a multidimensional grid and each ω_i is associated with one cell in this grid. The problem is to determine the set of parameters $\Omega^* \in \mathbb{R}^M$ such that

$$\Omega^* = \arg\max_{\Omega \in \mathbb{R}^M} \mathrm{U}(Z, T^*; S) \tag{5.5}$$

where T^* is the optimal threshold for a split of S using feature Z. The difference between the univariate and multivariate problems is the number of parameters to determine; there are m parameters in the univariate case and M parameters in the multivariate case, where $M \gg m$.

5.2 Univariate transformations

This section focuses on the univariate transformation problem which involes the creation of a new feature Z based on a numerical attribute Y and a single nominal attribute X. The presentation is divided into five sub-sections. The first one presents the approach for the evaluation of candidate solutions. The second section characterizes the problem space while the third section introduces the exhaustive solution. The last two sections describe heuristics developed to increase efficiency.

5.2.1 Evaluating candidate solutions

The optimization problem described by Equations (5.3) and (5.5) requires evaluation of the relevance function $\mathrm{U}(\Gamma(X, Y; \Omega), T^*; S)$ for many possible sets of parameters Ω. To achieve an appropriate level of performance, each potential set of parameters must be evaluated efficiently. This section introduces the approach we have developed to answer this need. This approach relies on the relevance matrices generated during the analysis of relevance (Section 4.2.4).

Let us recall that the usefulness of an attribute $Z = \Gamma(X, Y; \Omega)$, noted $\mathrm{U}(Z, T^*; S)$, is determined by the cumulative and balance sums of weights when considering all instances in S. These sums of weights are stored in the global relevance matrix for Z, noted RM_Z. As explained in Section 4.2.4, computing this global relevance matrix involes sorting of all instances in S and a global loop over the sorted instances to determine the cumulative and balance sums of weights. The sorting operation is time-consuming and becomes the bottleneck that limits the number of sets of parameters that can be investigated in practice. To resolve this problem, we have developed an approach that avoids the sorting operation. This approach efficiently generates RM_Z from the local relevance matrices already computed for Y (during the analysis of relevance). There are three steps. First, we obtain the local relevance matrices for Z given the ones already computed for Y. Second, we compute the global relevance matrix RM_Z from the local relevance matrices for Z. Finally, we compute $\mathrm{U}(Z, T^*; S)$. We now further detail these steps.

Computing the local relevance matrices for Z

Consider $S_1, S_2, \ldots, S_m$ a partition of S and let RM_i be the relevance matrix of Y for subset S_i with $i \in \{1, 2, \ldots, m\}$. Suppose we are investigating the usefulness of a feature $Z = \Gamma(X, Y; \Omega)$, with $\Omega = \{\omega_1, \omega_2, \ldots, \omega_m\}$, and let RM'_i be the local relevance matrices of Z in subset S_i. Following Equation (5.2), we have $Z = Y + \omega_i$ for all observations in subset S_i. Therefore, Y and Z have the same number of threshold values and the ordering of the instances is preserved (i.e., a sorting of this subset on Z or Y would produce the same sequence). Accordingly, the relevance matrices RM_i and RM'_i have the same number of entries and each of them has the same cumulative and balance sums of weights (Equation (4.11)) but the threshold values differ by ω_i. In other words, the local relevance matrices for Z and Y are identical except that the threshold values in RM'_i have been increased by ω_i.

Example 5 Suppose we work with the dataset listed in Table 5.1 and we want to transform Y based on X. The analysis of relevance for the effect of X on Y produces the two relevance matrices shown in Table 5.2. If we define a feature $Z = \Gamma(X, Y; \{5, 16\})$ (i.e., $\omega_1 = 5$, and $\omega_2 = 16$), then the relevance matrices for Z would be as shown in Table 5.3. As expected, RM'_1 is identical to RM_1 except that its threshold values have been increased by 5. Similarly, RM'_2 is like RM_2 but its threshold values have been increased by 16. □

Computing the global relevance matrix for Z

Figure 5.2 presents the algorithm `ComputeGlobalRM` that we propose to create the global relevance matrix from the local relevance matrices. To explain its behavior, we suppose that we want to generate RM_Y given the local relevance matrices $\mathrm{RM}_1, \mathrm{RM}_2, \ldots, \mathrm{RM}_m$.

The algorithm `ComputeGlobalRM` defines an entry in RM_Y for each observed value of Y in S. The list of observed values is given by the union of the threshold values in the local relevance matrices. The algorithm loops over all entries to compute their cumulative and balance sums of weights. For each entry, the algorithm first locates the *corresponding* cut points across the local relevance matrices. The corresponding cut point from a given RM_i is the cut point with the maximal threshold value that is smaller than or equal to the threshold value of the current entry (in the global matrix). If all thresholds in a given RM_i are higher than the current global threshold, then this RM_i does not contribute to the current entry in the global matrix. In the inner loop, the algorithm computes the cumulative sums of weights for the current entry by adding the cumulative sums of weights from all corresponding cut points. Finally, the algorithm obtains the balance sums of weights by subtracting the cumulative sums of weights from the global ones.

Obs	X	Y	class
1	a	5.0	1
2	a	5.0	1
3	a	10.0	1
4	a	10.0	1
5	a	10.0	2
6	a	15.0	1
7	a	15.0	1
8	a	20.0	1
9	a	20.0	2
10	a	20.0	2
11	a	20.0	2
12	a	20.0	2
13	b	2.0	1
14	b	4.0	1
15	b	4.0	1
16	b	4.0	1
17	b	10.0	2
18	b	10.0	2

Table 5.1: Dataset for Example 5. The first column is the observation number, X is the explanatory attribute, Y is the relevance attribute, and the last column is the class value. We assume that all instances have a weight of 1.0.

	RM_1		RM_2		
Thresh. values	Cumulative sums of weights $\{\|L_{1,1}\|, \|L_{1,2}\|\}$	Balance sums of weights $\{\|R_{1,1}\|, \|R_{1,2}\|\}$	Thresh. values	Cumulative sums of weights $\{\|L_{2,1}\|, \|L_{2,2}\|\}$	Balance sums of weights $\{\|R_{2,1}\|, \|R_{2,2}\|\}$
5	{2, 0}	{4, 5}	2	{1, 0}	{3, 2}
10	{4, 1}	{2, 4}	4	{4, 0}	{0, 2}
15	{6, 1}	{0, 4}	10	{4, 2}	{0, 0}
20	{6, 5}	{0, 0}			

Table 5.2: The two relevance matrices to represent the effect of X on Y.

RM'_1		
Thresh. values	Cumulative sums of weights $\{\|L_{1,1}\|, \|L_{1,2}\|\}$	Balance sums of weights $\{\|R_{1,1}\|, \|R_{1,2}\|\}$
10	{2, 0}	{4, 5}
15	{4, 1}	{2, 4}
20	{6, 1}	{0, 4}
25	{6, 5}	{0, 0}

RM'_2		
Thresh. values	Cumulative sums of weights $\{\|L_{2,1}\|, \|L_{2,2}\|\}$	Balance sums of weights $\{\|R_{2,1}\|, \|R_{2,2}\|\}$
18	{1, 0}	{3, 2}
20	{4, 0}	{0, 2}
26	{4, 2}	{0, 0}

Table 5.3: Resulting relevance matrices for $Z = \Gamma(X, Y; \{5, 16\})$. These matrices are similar to the corresponding initial ones (Table 5.2) except that the thresholds values have changes.

RM_Y			
Thresh. values	Cumulative sums of weights $\{\|L_{1,1}\|, \|L_{1,2}\|\}$	Balance sums of weights $\{\|R_{1,1}\|, \|R_{1,2}\|\}$	U
2	{1, 0}	{9, 7}	0.168
4	{4, 0}	{6, 7}	0.297
5	{6, 0}	{4, 7}	0.413
10	{8, 3}	{2, 4}	0.136
15	{10, 3}	{0, 4}	**0.507**
20	{10, 7}	{0, 0}	0.023

Table 5.4: The global relevance matrix for attribute Y as constructed using the algorithm `ComputeGlobalRM` and the locale relevance matrices RM_1 and RM_2 (Table 5.2. The righmost column shows the relevance of each cut point.

Figure 5.3 illustrates the identification of the corresponding cut points and the summation of the cumulative sums of weights using attribute Y from the dataset in Example 5. For each entry in RM_Y, the figure shows the corresponding cut points and the resulting cumulative sums of weights. Applying the algorithm `ComputeGlobalRM` for Z and Y, in the same example, generates the global relevance matrices shown in Tables 5.4 and 5.5, respectively. RM_Y has been obtained by combining the information from RM_1 and RM_2 (Table 5.2) while the RM_Z comes from RM'_1 and RM'_2 (Table 5.3).

Algorithm ComputeGlobalRM

Input: The set of relevance matrices $\mathrm{RM}_1, \mathrm{RM}_2, \ldots, \mathrm{RM}_m$ storing relevance info for a given attribute Y over subsets $S_1, S_2, \ldots, S_m$, respectively.

Output: The global relevance matrix RM storing the relevance info over the full dataset S.

Let **T** be the sequence of observed values for Y in S
$W_i \leftarrow |\{s \in S \mid \mathrm{val}_C(s) = c_i\}|^w$ for $i = 1, \ldots, k$ /* See Section 4.2.4 for def. of $|S|^w$ */
For each threshold $T \in \mathbf{T}$
    Add an entry in RM with threshold value T
    $L_i \leftarrow 0$ for $i = 1, \ldots, k$
    For $i = 1$ to m
        If RM_i has a threshold $\leq T$
            $T' \leftarrow$ the largest threshold in RM_i that is $\leq T$
            /* Assertion: T' is the corresponding cut point of T from RM_i */
            For $j = 1$ to k
                $L_j \leftarrow L_j +$ the j^{th} cumulative sum of weights in RM_i at threshold T'
    For $j = 1$ to k
        $R_j \leftarrow W_j - L_j$
    Set the cumulative sums of weights in RM at T to $\{L_1, L_2, \ldots, L_k\}$
    Set the balance sums of weights in RM at T to $\{R_1, R_2, \ldots, R_k\}$
return RM

Figure 5.2: Algorithm `ComputeGlobalRM` to compute the global relevant matrix given the relevant matrices $\mathrm{RM}_1, \mathrm{RM}_2, \ldots, \mathrm{RM}_m$ associated with the subsets $S_1, S_2, \ldots, S_m$, respectively.

Computing the relevance of Z

The final step determines the usefulness of the given transformation by computing $\mathrm{U}(Z, T^*; S)$. Our approach is to compute the relevance function $\mathrm{U}(Z, T; S)$ for all thresholds T in the global relevance matrix RM_Z. The threshold with the maximal relevance value is selected as the best threshold and its relevance accounts for the global relevance of Z. These compu-

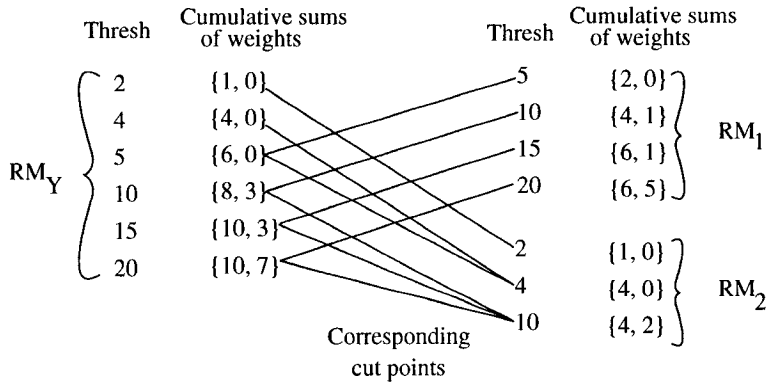


Figure 5.3: Identification of the corresponding cut points across local relevance matrices for all entries in the global relevance matrix for Y (Example 5).

RM_Z			
Thresh. values	Cumulative sums of weights $\{\|L_{1,1}\|, \|L_{1,2}\|\}$	Balance sums of weights $\{\|R_{1,1}\|, \|R_{1,2}\|\}$	U
10	{2, 0}	{8, 7}	0.210
15	{4, 1}	{6, 6}	0.090
18	{5, 1}	{5, 6}	0.134
20	{10, 1}	{0, 6}	**0.762**
25	{10, 5}	{0, 2}	0.342
26	{10, 7}	{0, 0}	0.023

Table 5.5: The global relevance matrix for attribute Z as constructed using the algorithm `ComputeGlobalRM` and the relevance matrices RM'_1 and RM'_2 (Table 5.3). The righmost column shows the relevance of each cut point.

tations are performed very efficiently since all the required information is readily available in RM_Z. However, it is possible to optimize the selection of the best threshold by considering only the cut points identified as *boundary points* [Fayyad & Irani, 1992; Codrington & Brodley, 1999; Elomaa & Rousu, 1999]. Our current implementation does not use the optimization based on boundary points since we need the relevance values for all cut points when we generate the relevance graphs.

The right most column in Tables 5.4 and 5.5 provides the relevance values for all cut points for Y and Z, respectively. The maximal relevance value for Y is 0.507, which is achieved at threshold 15 (i.e., $T^* = 15$). After applying the transformation $\Gamma(X, Y; \{5, 16\})$, we obtain a feature Z with a global relevance of 0.762, at threshold 20. This increase in global relevance shows the usefulness of the candidate solution $\Omega = \{5, 16\}$ for the dataset considered.

In summary, this section presented an approach to evaluate a candidate transformation $\Gamma(X, Y; \Omega)$ by using only the information computed during the analysis of relevance (i.e., the relevance matrices). Using this approach, we can quickly determine the usefulness of any candidate set of parameters Ω. We now turn to the analysis of the sets of parameters that must be investigated to ensure an optimal solution.

5.2.2 The problem space

The objective is to find $\Omega^* \in \mathbb{R}^m$ that maximizes $\mathrm{U}(Z, T^*; S)$ (Equation (5.3)). As mentioned earlier, there is an infinite number of candidate sets of parameters $\Omega \in \mathbb{R}^m$. On the other hand, many of these sets of parameters can be considered as duplicate since they lead to features Z with identical relevance information. By avoiding these duplicates, we can reduce the problem to a finite search space. We first explain how to identify the duplicate sets of parameters and then present the size of the problem space.

We have seen that $\mathrm{U}(Z, T^*; \Omega)$ depends on the cumulative and balance sums of weights stored in the last two columns of RM_Z. We use the values in these two columns to represent a state in the solution space, noted A. Following algorithm `ComputeGlobalRM` (Figure 5.2), the values in A are determined by the weights of the corresponding cut points across the local relevance matrices. Therefore, all sets of parameters that lead to the same corresponding cut points for all entries in the global relevance matrix would also lead to the same state. To obtain an optimal solution, we only need to visit each state once and hence, it is sufficient to consider only one of these sets of parameters.

To illustrate let us consider the example from the previous section. The set of parameters used to generate Z was $\Omega = \{5, 16\}$ which lead to the state given by the values in the second and third columns of the relevance matrix in Table 5.5. Figure 5.4 (a) shows the corresponding cut points across RM_1 and RM_2 associated with this state. On the other hand, $\Omega = \{0, 11\}$ also produces the same corresponding cut points for all entries in the global relevance matrix (Figure 5.4 (b)). Consequently, both sets of parameters ($\Omega = \{5, 16\}$ and $\Omega = \{0, 11\}$) lead to the same state. Other examples of sets of parameters that lead to the same state are: $\{1, 12\}$, $\{-5, 6\}$, and all sets $\{5 * b, 16 * b\}$ with $b \in \mathbb{R}$. The global

relevance matrices produced by these various sets of parameters are all identical to RM_Z except for the threshold values. This means that all the resulting features cut the instance space in the same manner. For learning, all of these features are equivalent. Consequently, it is sufficient to consider only one of these sets of parameters.

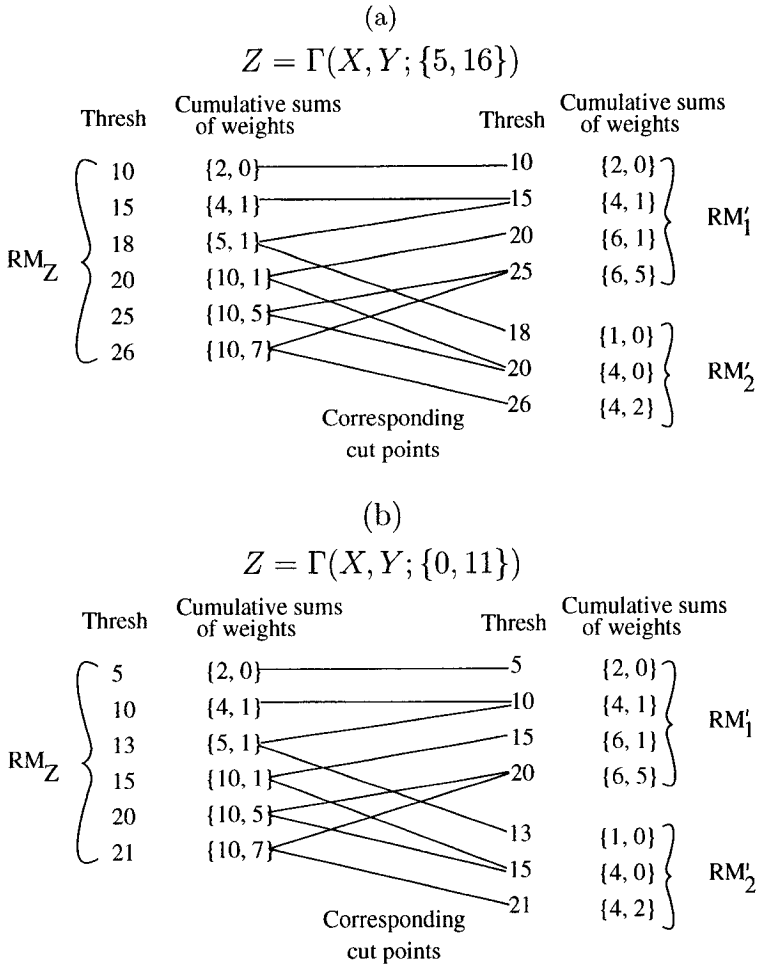


Figure 5.4: Corresponding cut points across local relevance matrices for all entries in the global relevance matrix for $Z = \Gamma(X, Y; \{5, 16\})$ and $Z = \Gamma(X, Y; \{0, 11\})$. We note that these transformations have identical corresponding cut points.

To avoid duplicated states, we need sets of parameters that change the alignment of the corresponding cut points across the local relevance matrices $\mathrm{RM}_1, \mathrm{RM}_2, \ldots, \mathrm{RM}_m$. The number of distinct alignments of the corresponding cut points across the subsets is given by

$\prod_{i=1}^{m} d_i$ where m is the number of subsets in the partition and d_i is the number of distinct values in the i^{th} subset. Each of these alignments results in a unique state A. To perform an exhaustive search, we need to evaluate each of these states. Before presenting the algorithm to perform such a search, let us note that the number of states is exponential in the number of subsets. This explains why we were mentioning in Section 4.2.3 that it is preferable to minimize the number of bins when discretizing explanatory numerical attributes.

5.2.3 The exhaustive approach

Figure 5.5 presents the detailed algorithm for the exhaustive solution. This algorithm, named `UnivExhaustiveSearch`, finds the best set of parameter values for Ω by evaluating all possible states. For each state, the algorithm performs three steps. First, it computes a set of parameter values Ω associated with the current state. Second, the algorithm evaluates the usefulness of the candidate solution. Third, it compares the current candidate solution with the previous best one and updates its best solution as needed.

For the first state (or first iteration), the algorithm sets ω_1 to the first observed value in S_1, ω_2 to the first observed value in S_2, ..., and ω_m to the first observed value in S_m. For the second state, $\omega_1, \omega_2, \ldots, \omega_{m-1}$ stay the same but ω_m takes the second observed value in S_m. For the third state, ω_m takes the third observed value in S_m and others keep the same values. The process continues until all possible combinations of threshold values are considered. Each candidate solution is evaluated using the approach described in Section 5.2.1.

For the dataset introduced in Example 5, the total number of possible states is $d_1 * d_2 = 4 * 3 = 12$. The sets of Ω used to evaluate these 12 states are listed in the second column of Table 5.6. The third column provides the relevance for the best cut point for transformation using the specified Ω. Table 5.7 further details the partial results by showing the 12 global relevance matrices corresponding to the 12 states.

We first notice that the 12 sets of parameters lead to 12 different states. This is verified through Table 5.7 since the cumulative and balance sums of weights differ for at least one threshold from one global relevance matrix to another. Second, we observe that the transformation discussed earlier $Z = \Gamma(X, Y; \{5, 16\})$ is equivalent to the 8^{th} transformation $\Omega = \{-15, -4\}$ since the cumulative and balance sums of weights for this transformation are identical to the ones shown in Table 5.5.

Finally, we note that the maximal relevance value 0.762 is achieved in five of the 12 different states for this problem. These five states happen to share a global cut point for which the cumulative sums of weights are $\{10, 1\}$, just like the point with threshold value

Algorithm UnivExhaustiveSearch

Input: The set of relevance matrices $\mathrm{RM}_1, \mathrm{RM}_2, \ldots, \mathrm{RM}_m$ storing relevance info for a given attribute Y over subsets $S_1, S_2, \ldots, S_m$, respectively.

Output: A set of constants Ω^* that maximizes $\mathrm{U}(\Gamma(X, Y; \Omega), T^*; S)$.

$bestScore \leftarrow \mathrm{U}(Y, T^*; S)$
$\Omega^* \leftarrow \{0, \ldots, 0\}$
$d_0 \leftarrow 1$
$numStates \leftarrow 1$
For $i = 1$ to m
    $d_i \leftarrow$ the number of distinct values for Y in S_i
    $numStates \leftarrow numStates * d_i$
For $i = 1$ to $numStates$
    $stateId \leftarrow i - 1$
    /* Set Ω for the current state */
    For $j = m$ to 1 by -1
        $curDivisor \leftarrow \prod_{k=0}^{j-1} d_k$
        $l \leftarrow \lfloor stateId / curDivisor \rfloor$
        $v \leftarrow$ the $(l+1)^{th}$ distinct value of Y in S_i
        $\omega_j \leftarrow 0 - v$
        $stateId \leftarrow stateId - (l * curDivisor)$
    $\Omega \leftarrow \{\omega_1, \omega_2, \ldots, \omega_m\}$
    /* Evaluate current state */
    Compute $\mathrm{RM}'_1, \mathrm{RM}'_2, \ldots, \mathrm{RM}'_m$ for $Z = \Gamma(X, Y; \Omega)$
    $\mathrm{RM}_Z \leftarrow$ `ComputeGlobalRM`$(\{\mathrm{RM}'_1, \mathrm{RM}'_2, \ldots, \mathrm{RM}'_m\})$
    $curScore \leftarrow \mathrm{U}(Z, T^*; S)$ /* Computed from RM_Z */
    /* Remember the best state */
    if $curScore > bestScore$
      $bestScore \leftarrow curScore$
      $\Omega^* \leftarrow \Omega$
return Ω^*

Figure 5.5: Algorithm `UnivExhaustiveSearch` to perform an exhaustive search to find the set of constants Ω^* that maximizes $\mathrm{U}(\Gamma(X, Y; \Omega), T^*; S)$.

State	Ω	$U(Z, T^*; \Omega)$
1	{-5, -2}	0.507
2	{-5, -4}	0.507
3	{-5, -10}	0.507
4	{-10, -2}	0.762
5	{-10, -4}	0.762
6	{-10, -10}	0.507
7	{-15, -2}	0.762
8	{-15, -4}	0.762
9	{-15, -10}	0.507
10	{-20, -2}	0.342
11	{-20, -4}	0.342
12	{-20, -10}	0.762

Table 5.6: The sets of parameters used to evaluate the 12 possible states with corresponding global relevance for the dataset in Example 5.

20 in Table 5.5. The only way to obtain such a global cut point with the given partition is through a transformation that aligns the second cut point in RM_1 and the third cut point in RM_2[1]. A further look at these two cut points reveals two important characteristics. First, they are compatible according to Theorem 1 (Section 4.3.2). Second, they are the ones with the highest relevance in their respective local relevance matrices. As we will see in the next section, such observations can be used to develop efficient heuristics as potential alternatives to the exhaustive approach.

5.2.4 Restricting the search space to most promising cut points

The exhaustive approach introduced above takes into account an exponential number of states. This complexity may limit the applicability of the approach in several situations. To address this problem, we have developed a heuristic that controls the complexity of the search by limiting the analysis to states that are likely to contain the optimal solution.

When the most relevant cut points from the various subsets are compatible, the optimal solution is to use a transformation that aligns these cut points. We have already applied this

[1] $Z = \Gamma(X, Y; \{5, 16\})$ and $Z = \Gamma(X, Y; \{15, -4\})$ are two examples of transformations that produce this alignment as shown in Figure 5.4.

State 1 ($\Omega = \{-5, -2\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
0	{3, 0}	{7, 7}	0.251
2	{6, 0}	{4, 7}	0.413
5	{8, 1}	{2, 6}	0.353
8	{8, 3}	{2, 4}	0.136
10	**{10, 3}**	**{0, 4}**	**0.507**
15	{10, 7}	{0, 0}	0.023

State 2 ($\Omega = \{-5, -4\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-2	{1, 0}	{9, 7}	0.168
0	{6, 0}	{4, 7}	0.413
5	{8, 1}	{2, 6}	0.353
6	{8, 3}	{2, 4}	0.136
10	**{10, 3}**	**{0, 4}**	**0.507**
15	{10, 7}	{0, 0}	0.023

State 3 ($\Omega = \{-5, -10\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-8	{1, 0}	{9, 7}	0.168
-6	{4, 0}	{6, 7}	0.297
0	{6, 2}	{4, 5}	0.094
5	{8, 3}	{2, 4}	0.136
10	**{10, 3}**	**{0, 4}**	**0.507**
15	{10, 7}	{0, 0}	0.023

State 4 ($\Omega = \{-10, -2\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-5	{2, 0}	{8, 7}	0.210
0	{5, 1}	{5, 6}	0.134
2	{8, 1}	{2, 6}	0.353
5	**{10, 1}**	**{0, 6}**	**0.762**
8	{10, 3}	{0, 4}	0.507
10	{10, 7}	{0, 0}	0.023

State 5 ($\Omega = \{-10, -4\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-5	{2, 0}	{8, 7}	0.210
-2	{3, 0}	{7, 7}	0.251
0	{8, 1}	{2, 6}	0.353
5	**{10, 1}**	**{0, 6}**	**0.762**
6	{10, 3}	{0, 4}	0.507
10	{10, 7}	{0, 0}	0.023

State 6 ($\Omega = \{-10, -10\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-8	{1, 0}	{9, 7}	0.168
-6	{4, 0}	{6, 7}	0.297
-5	{6, 0}	{4, 7}	0.413
0	{8, 3}	{2, 4}	0.136
5	**{10, 3}**	**{0, 4}**	**0.507**
10	{10, 7}	{0, 0}	0.023

State 7 ($\Omega = \{-15, -2\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-10	{2, 0}	{8, 7}	0.210
-5	{4, 1}	{6, 6}	0.090
0	{7, 1}	{3, 6}	0.259
2	**{10, 1}**	**{0, 6}**	**0.762**
5	{10, 5}	{0, 2}	0.342
8	{10, 7}	{0, 0}	0.023

State 8 ($\Omega = \{-15, -4\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-10	{2, 0}	{8, 7}	0.210
-5	{4, 1}	{6, 6}	0.090
-2	{5, 1}	{5, 6}	0.134
0	**{10, 1}**	**{0, 6}**	**0.762**
5	{10, 5}	{0, 2}	0.342
6	{10, 7}	{0, 0}	0.023

State 9 ($\Omega = \{-15, -10\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-10	{2, 0}	{8, 7}	0.210
-8	{3, 0}	{7, 7}	0.251
-6	{6, 0}	{4, 7}	0.413
-5	{8, 1}	{2, 6}	0.353
0	**{10, 3}**	**{0, 4}**	**0.507**
5	{10, 7}	{0, 0}	0.023

State 10 ($\Omega = \{-20, -2\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-15	{2, 0}	{8, 7}	0.210
-10	{4, 1}	{6, 6}	0.090
-5	{6, 1}	{4, 6}	0.189
0	{7, 5}	{3, 2}	0.023
2	**{10, 5}**	**{0, 2}**	**0.342**
8	{10, 7}	{0, 0}	0.023

State 11 ($\Omega = \{-20, -4\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-15	{2, 0}	{8, 7}	0.210
-10	{4, 1}	{6, 6}	0.090
-5	{6, 1}	{4, 6}	0.189
-2	{7, 1}	{3, 6}	0.259
0	**{10, 5}**	**{0, 2}**	**0.342**
6	{10, 7}	{0, 0}	0.023

State 12 ($\Omega = \{-20, -10\}$)

T	Cum	Bal	U$(Z, T; \Omega)$
-15	{2, 0}	{8, 7}	0.210
-10	{4, 1}	{6, 6}	0.090
-8	{5, 1}	{5, 6}	0.134
-6	{8, 1}	{2, 6}	0.353
-5	**{10, 1}**	**{0, 6}**	**0.762**
0	{10, 7}	{0, 0}	0.023

Table 5.7: The global relevance matrices obtained during the exhaustive search for the best univariate transformation using the dataset from Example 5.

solution for two datasets. First, during the presentation of the interactive relevance graphs in which we used the interactive tool to manually align the most relevant cut points across the subsets. The dataset used in this example was introduced in Section 4.2.2. Figure 4.15 depicted the effect of the optimal transformation. The second example has been presented in the previous section while illustrating the exhaustive search algorithm. In both cases, we could have bypassed the search and picked the best solution directly since all of the most relevant cut points were compatible.

The situation is more complex when the most relevant cut points are not all compatible. The negative effect of incompatibility between highly relevant cut points is so important (Section 4.3.3) that we need to consider combinations of cut points that are not maximally relevant in their own subset. Our approach to cover the various possibilities is to include, from each subset, the most relevant cut points for all observed compatibility patterns expressed by the combined values of the two distribution characteristics λ_1 and λ_2 (Definition 4). As explained in Section 4.3.2, the number of possible values for these two characteristics combined is k^2, where k is the number of class values. In practice, we often observe only a subset of these potential combinations in a given subset. In that sense, k^2 is the upper bound on the number of cut points to consider from each subset.

There are two additional issues to consider while handling the case of incomplete compatibility. First, it may happen that one or both of the distribution characteristics are undetermined due to equal sums of weights between two or more classes (see remark in Definition 4). It is important to consider these cases as well since alignment with a cut point that has one or two undetermined distribution characteristics might be better than an alignment with any other cut points. We resolve this issue by adding up to three additional cut points per subset to cover the three possible situations (i.e., only $\lambda_1 =?$, only $\lambda_2 =?$, and $\lambda_1 = \lambda_2 =?$). The actual number of additional cut points may be less than three since not all subsets may have cut points that satisfy these three situations.

The second issue arises when the subsets have no compatible cut points. An example of this situation was presented in Figure 4.8. With such a partition, the best solution is to try to minimize the negative effects of the incompatibility by trying to align the minimally relevant cut points. As discussed in Section 4.3.1, these minimally relevant cut points are located at the extremes of the relevance curves. Consequently, we consider the alignment with the minimally relevant cut points by incorporating the cut points with minimal and maximal threshold. This adds a maximum of two cut points per subset. To avoid adding the same cut point twice, we need to make sure that these extremes have not been already considered (due to their distribution characteristics).

Figure 5.6 presents the detailed algorithm for the general case of non-compatibility between the maximally relevant cut points. The algorithm's name is `ReduceRM` (Reduce Relevance Matrix). The algorithm takes as input a relevance matrix. It identifies the cut points mentioned above and stores them into a new relevance matrix. At the end, the algorithm outputs the new reduced relevance matrix. The algorithm is easily integrated as a preprocessor to the algorithm `UnivExhaustiveSearch` (Figure 5.5). The idea is to call the algorithm `ReduceRM` for each local relevance matrix for Y and then pass the new simplified relevance matrices as input to the algorithm `UnivExhaustiveSearch`.

The use of the algorithm `ReduceRM` can greatly reduce the size of the search space. The maximal size of a simplified relevance matrix is $k^2 + 3 + 2$ (maximally relevant cut points + undetermined characteristics + extremes). Thus, with this algorithm, the search space will have at most $(k^2 + 5)^m$ states, where m is the number of subsets in the partition. Compared to the exhaustive search space, which has a complexity of $\prod_{i=1}^{k} d_i$, the reduction in the number of states could be very significant. The gain comes from the fact that k is typically very small (usually $k = 2$) which leads to $(k^2 + 5) < d_i$, where d_i is the number of distinct values for Y in S_i. Because of the multiplicative effect, we would typically have $(k^2 + 5)^m \ll \prod_{i=1}^{k} d_i$. It is also important to note that the algorithm `ReduceRM` always generates a relevance matrix that is smaller or has the same size as the original one.

To illustrate the potential benefit of the algorithm `ReduceRM`, let us suppose that the maximally relevant cut points for the dataset introduced in Section 4.2.2 were not all compatible. The partition for this example has three subsets having respectively 20, 42, and 38 distinct values. Therefore, the size of the exhaustive search space is $20 * 42 * 38 = 31920$. On the other hand, the algorithm `ReduceRM` would have selected exactly five points in each of the three subsets (two points for the extremes plus three points to cover the observed combinations of values for λ_1 and λ_2). This means that the size of the resulting search space would have been $5 * 5 * 5 = 125$ for a reduction of over 99.6%. Clearly, the reduction would not be as important in simple domains such as the one discussed in the previous section where the number of distinct values in each group is very small.

Finally, we have presented the algorithm `ReduceRM` as a heuristic since we have no proof of its optimality. On the other hand, we used this algorithm on various real world and artificial datasets and did not encounter any sub-optimal solution. Note that a proof of optimality would not be as satisfying as it may appear since there are situations in which the complexity reduction provided by the algorithm `ReduceRM` is not sufficient. In the next section, we discuss such situations and present the greedy solution we have adopted.

Algorithm ReduceRM
Input: A relevance matrix RM
Output: A reduced relevance matrix, noted RRM, that contains
only the most promising cut points.

$k \leftarrow$ number of class values
RRM $\leftarrow$ a new empty relevance matrix
/* Adding maximally relevant cut points */
For $i = 1$ to k
  For $j = 1$ to k
    if RM has at least one entry with $\lambda_1 = i$ and $\lambda_2 = j$
      index $\leftarrow$ index of the most relevant cut point in RM with $\lambda_1 = i$ and $\lambda_2 = j$
      Add RM[*index*] to RRM
/* Adding cut points with undetermined λ_1 and/or λ_2 as needed*/
if RM has at least one entry with $\lambda_1 =?$ and $\lambda_2 \neq ?$
  index $\leftarrow$ index of the most relevant cut point in RM with $\lambda_1 =?$ and $\lambda_2 \neq ?$
  Add RM[*index*] to RRM
if RM has at least one entry with $\lambda_1 \neq ?$ and $\lambda_2 =?$
  index $\leftarrow$ index of the most relevant cut point in RM with $\lambda_1 \neq ?$ and $\lambda_2 =?$
  Add RM[*index*] to RRM
if RM has at least one entry with $\lambda_1 =?$ and $\lambda_2 =?$
  index $\leftarrow$ index of the most relevant cut point in RM with $\lambda_1 =?$ and $\lambda_2 =?$
  Add RM[*index*] to RRM
/* Adding extremas as needed */
if RM[1] not in RRM
  Add RM[1] to RRM
last $\leftarrow$ the number of entries in RM
if RM[*last*] not in RRM
  Add RM[*last*] to RRM
return RRM

Figure 5.6: Algorithm `ReduceRM` which removes from a given relevance matrix all cut points that are not likely to lead to an optimal solution.

5.2.5 A greedy search: 2 partitions at a time

The algorithm `ReduceRM` limits the complexity of the search by reducing the number of points that need to be considered from each subset of the partition. Due to the multiplicative effect, this could lead to great performance improvement. On the other hand, when the partitioning is based on an explanatory attribute that has many values, the resulting complexity may still be too great. For instance, let us suppose we want to cancel the effect of `car_model` (the explanatory attribute) on the global relevance of `fuel_consumption` (the response attribute). If there are, let's say, 50 kinds of car models in the dataset, then even if we need to consider only 5 thresholds per subset, the size of the search space would be 5^{50} which is still considerable. We address this problem through a greedy search algorithm that considers only two partitions at a time. Figure 5.7 presents the algorithm, named `UnivBy2ThenReduceSearch`.

This algorithm is an alternative to the algorithm `UnivExhaustiveSearch` (Figure 5.5). Both find a set of parameter values for Ω given the local relevance matrices for Y. However, there are two fundamental differences between the two algorithms. First, the search method `UnivExhaustiveSearch` performs an exhaustive search while `UnivBy2ThenReduceSearch` uses a greedy search strategy. The latter search strategy considers only two subsets at a time and keeps the best partial results for the following iterations. Second, the algorithm `UnivBy2ThenReduceSearch` relies on the optimization procedures described in the previous section. These include the computation of the optimal solution when the maximally relevant cut points are compatible and the use of the algorithm `ReduceRM` to reduce the complexity of the task in each iteration.

The greedy search strategy reduces the complexity by an order of magnitude. Precisely, if we assume that all subsets have d distinct values, then the greedy search would have a complexity of $(m-1)d^2$ by comparison to d^m for the exhaustive solution. Moreover, the term d^2 is further reduced to a relatively small constant by using the algorithm `ReduceRM` at the beginning of the process and between each iteration. Consequently, the complexity of the algorithm `UnivBy2ThenReduceSearch` is quasi-linear in the number of subsets in the partition. On the other hand, this simplification goes with the risk of finding a non-optimal solution for Ω. In turn, a non-optimal solution may limit the usefulness of the proposed solution. We address this concern in Chapter 6 through an empirical evaluation of the feasibility of the proposed approach. Before introducing this evaluation, lets us consider the case of multivariate transformations.

Algorithm UnivBy2ThenReduceSearch

Input: The set of relevance matrices $\mathrm{RM}_1, \mathrm{RM}_2, \ldots, \mathrm{RM}_m$ storing relevance info for a given attribute Y over subsets $S_1, S_2, \ldots, S_m$, respectively.

Output: A set of constants Ω^* that maximizes $\mathrm{U}(\Gamma(X, Y; \Omega), T^*; S)$.

```
/* Check for maximally relevant compatible cut points */
if λ1(Y, T*; S_i) = λ1(Y, T*; S_j) and λ2(Y, T*; S_i) = λ2(Y, T*; S_j) for i ≠ j
  {T*_1, T*_2, ..., T*_m} ← best cut point thresholds in {RM_1, RM_2, ..., RM_m}
  Ω* ← {−T*_1, −T*_2, ..., −T*_m}
  return Ω*

/* Pre-processing local relevance matrices */
for i = 1 to m
    RM_i ← ReduceRM(RM_i)
RM_cum ← RM_1
For i = 2 to m
    /* Find solution for ω_{i−1} and ω_i */
    {ω_{i−1}, ω_i} ← UnivExhaustiveSearch(RM_cum, RM_i)
    /* Update global relevance matrix based on partial solution */
    Compute RM'_{i−1}, RM'_i based on Ω = {ω_{i−1}, ω_i}
    RM_cum ← ComputeGlobalRM({RM'_{i−1}, RM'_i})
    /* Reduce current global relevance matrix */
    RM_cum ← ReduceRM(RM_cum)
Ω* ← {ω_1, ω_2, ..., ω_m}
return Ω*
```

Figure 5.7: Algorithm `UnivBy2ThenReduceSearch` which performs a search to find the set of constants Ω^* that maximizes $\mathrm{U}(\Gamma(X, Y; \Omega), T^*; S)$ by considering only 2 subsets at a time. The algorithm also uses `ReduceRM` to simplify the task in each iteration.

5.3 Multivariate transformations

A multivariate transformation involves a response attribute Y and l explanatory attributes noted $\mathbf{X} = \{X_1, X_2, \ldots, X_l\}$. The objective is to find a set of values for Ω such that the global relevance of a new feature $Z = \Gamma(\mathbf{X}, Y; \Omega)$ is maximized (Equations (5.4) and (5.5)). In this case, Ω has $M = \prod_{i=1}^{l} m_i$ values where $m_i = |\text{Dom}(X_i)|$ for $i = 1, 2, \ldots, l$. We describe two approaches to resolve the multivariate problem. The first one is a direct extension of the univariate solution. The second is an inductive approach that scales better to high dimensional relationships. The next section describes the extension of the univariate solution and the following four sections focus on the inductive approach.

5.3.1 Extension of the univariate approach

The univariate procedure incorporates the information from the explanatory attribute at the very beginning of the process, that is, during the partitioning operation. The univariate partitioning creates one subset for each observed value of the explanatory attribute. Then, for each of these subsets, the procedure creates a local relevance matrix and computes its corresponding ω value. To generalize the univariate solution to the multivariate case, we need to take into account the joint effects of several explanatory attributes $\{X_1, X_2, \ldots, X_l\}$. We do this through a multivariate partitioning operation that generates subsets for all possible combinations of values for $\{X_1, X_2, \ldots, X_l\}$. Precisely,

Definition 6 A *multivariate partitioning* of a dataset S involving l explanatory attributes $X_1, \ldots, X_l$, with $|\text{Dom}(X_i)| = m_i$ for $i = \{1, 2, \ldots, l\}$, generates a collection $S_1, S_2, \ldots, S_M$ of subsets such that

$$
\begin{aligned}
S_1 &= \{s \in S \mid X_i = X_i(1)\} \text{ for } i = 1, 2, \ldots, l \\
S_2 &= \{s \in S \mid X_1 = X_1(2), X_i = X_i(1)\} \text{ for } i = 2, 3, \ldots, l \\
&\vdots \\
S_{m_1} &= \{s \in S \mid X_1 = X_1(m_1), X_i = X_i(1)\} \text{ for } i = 2, 3, \ldots, l \\
S_{m_1+1} &= \{s \in S \mid X_2 = X_2(2), X_i(s) = X_i(1)\} \text{ for } i = 1, 3, 4, \ldots, l \\
S_{m_1+2} &= \{s \in S \mid X_1 = X_1(2), X_2 = X_2(2), X_i = X_i(1)\} \text{ for } i = 3, 4, \ldots, l \\
&\vdots \\
S_M &= \{s \in S \mid X_i = X_i(m_i)\} \text{ for } i = 1, 2, \ldots, l
\end{aligned}
$$

Remark 2 We note that a *multivariate partitioning* does not necessarily produce a partition (Definition 1) since the subsets are not guaranteed to be non-empty. □

Using these subsets, we can proceed exactly as with the univariate approach (i.e., create a relevance matrix for each subset and then compute its corresponding ω value using the univariate solution).

The multivariate partitioning is a simple method that is useful when the number of possible combinations of values (M) is not too high. For instance, when $M \leq 20$, we can use this approach with the relevance graphs to quickly assess the combined effect of more than one explanatory attribute. As M increases above this number, the readability of the relevance graph tends to decrease rapidly. In terms of the automatic search for the transformation parameters, the algorithm `UnivBy2ThenReduceSearch` could tolerate a larger M. For instance, our non-optimized Java implementation of this algorithm would generally produce a solution within one minute with $M \approx 1000$.

However, we have to recognize that the number of possible combinations of values (and subsets) grows exponentially in the number of explanatory attributes. This is problematic for two reasons. First, having a finite dataset and a large number of subsets means that each of them is likely to contain very few instances. In turn, few instances make it difficult to perform a reliable analysis. The second issue with the multivariate partitioning is the growth of complexity. Even if we have sufficient data in each subset, analyzing an exponential number of subsets is generally impractical, however efficient is the analysis. Therefore, when we are interested in the combined effects of explanatory attributes having multiple values, we need an alternative approach that does not oversplit the initial dataset. The following section presents such an approach.

5.3.2 An inductive approach

Our second solution learns the multivariate transformation models through an inductive process. We apply this process for each response attribute independently. Figure 5.8 illustrates this inductive approach. The input to this process is the set of univariate transformation models for the given response attribute. The processing may extend over several phases. Each phase has two steps: *Feature Generation* and *Feature Selection*. The Feature Generation step creates new features by combining the ones obtained previously. The Feature Selection step evaluates the newly created features and discards the ones that do not improve performance. The output includes all transformations that have been selected over the various phases. As shown in Figure 5.8, we use the notation Z_i^j to represent the i^{th}

multivariate feature in phase j. Similarly, we shall use Ω_i^j to represent the parameter values for Z_i^j (i.e., $Z_i^j = \Gamma(\mathbf{X}, Y; \Omega_i^j)$). As in the previous sections, the univariate features do not have a superscript. The following two sections elaborate on the Feature Generation and Feature Selection steps, respectively. The last section discusses the stopping criteria as well as other arguments that can be used to introduce more flexibility.

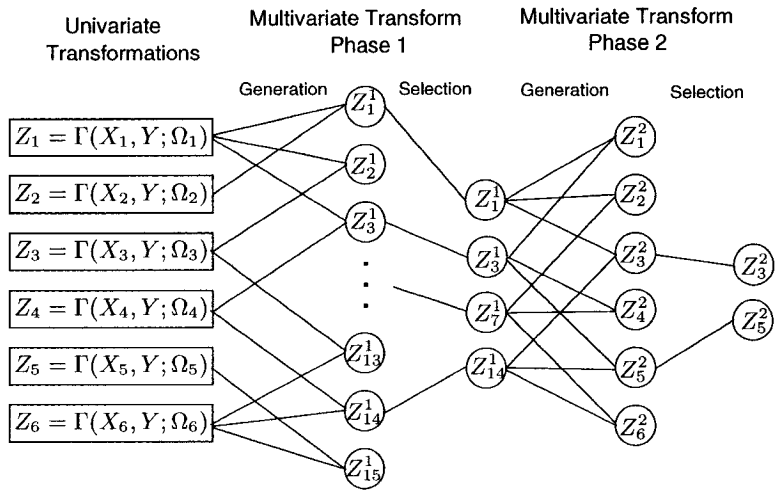


Figure 5.8: Overview of the inductive approach to generate multivariate transformations.

5.3.3 Mutlivariate Feature Generation

The generation step constructs features in progressive order of complexity. In the first phase, it uses the initial univariate transformations to create multivariate features with two explanatory attributes. For instance, the feature Z_1^1 in Figure 5.8 comes from the univariate features Z_1 and Z_2, which are based on the explanatory attributes X_1 and X_2, respectively. Therefore, we have $Z_1^1 = \Gamma(\{X_1, X_2\}, Y; \Omega_1^1)$. In the second phase, the generation step uses the selected features from the first phase to create features involving either three or four explanatory attributes. For example, feature Z_1^2 is based on three explanatory attributes (X_1, X_2, and X_4) while feature Z_3^2 involves four explanatory attributes (X_1, X_2, X_4, and X_6). The complete definitions for these two features are $Z_1^2 = \Gamma(\{X_1, X_2, X_4\}, Y; \Omega_1^2)$ and $Z_3^2 = \Gamma(\{X_1, X_2, X_4, X_6\}, Y; \Omega_3^2)$. In the third phase, the new features engage between five and eight explanatory attributes, and so forth. In each phase, the generation step creates multivariate features for all possible pairs of features from the previous phase. Therefore, if the previous phase has r features, then the generation step for the current phase would

generate C_r^2 features.

Each feature is constructed through an iterative optimization process. In the first iteration, a new multivariate feature Z involving l explanatory attributes $X_1, \ldots, X_l$ is defined as[2]

$$\begin{aligned} Z^0 &= \Gamma(\{X_1, X_2, \ldots, X_l\}, Y; \Omega^0, S') \\ &= \Gamma(X_1, Y; \Omega_1^0, S') \\ &\quad + \Gamma(X_2, Y; \Omega_2^0, S') + \ldots \\ &\quad + \Gamma(X_l, Y; \Omega_l^0, S') - (l-1) * Y \end{aligned} \tag{5.6}$$

where each Ω_i^0 is the set of parameter values that optimizes $\mathrm{U}(\Gamma(X_i, Y; \Omega, S'), T^*; S')$ for $i = \{1, 2, \ldots, l\}$, and S' is a subset of the training set. As discussed in the next section, the remaining part of the training set is kept for the selection phase. The last term in the equation is required to cancel the effect of the repeated additions of the initial Y value (Equation (5.2)).

The following iterations progressively refine the sets of parameter values using the following rule:

$$\Omega_i^j = \arg\max_{\Omega \in \mathbb{R}^{m_i}} \mathrm{U}(\Gamma(X_i, Z^{j-1}; \Omega_i^{j-1}), T^*; S) \tag{5.7}$$

where $j \geq 1$ is the iteration number and $i = \{1, 2, \ldots, l\}$. The procedure uses these new sets of parameter values to update the new feature

$$\begin{aligned} Z^j &= \Gamma(X_1, Z^{j-1}; \Omega_1^j, S') \\ &\quad + \Gamma(X_2, Z^{j-1}; \Omega_2^j, S') + \ldots \\ &\quad + \Gamma(X_l, Z^{j-1}; \Omega_l^j, S') - (l-1) * Y \end{aligned} \tag{5.8}$$

The final step in each iteration is to compute the global relevance of the new feature $\mathrm{U}(Z^j, T^*; S')$ and compare it with the previous one $\mathrm{U}(Z^{j-1}, T^*; S')$. If the difference between the two iterations is smaller than a pre-defined ϵ, the process stops and returns the best sets of parameter values found.

[2]The superscript in this equation refers to the iteration number while optimizing the parameters for the given feature. This is different than the superscripts used in Figure 5.8 to distinguish features from various phases.

5.3.4 Mutlivariate Feature Selection

The feature generation step tries to maximize the global relevance through generation of potentially very complex transformations. There is a risk that these models overfit[3] the data. To avoid the overfitting problem, we use only a portion of the training data for feature generation and use the remaining part for selection. These two subsets are noted S' and S'', respectively. S'' is also named the *validation* dataset. The feature selection evaluates the relevance of all new features produced by the feature generation step in the same phase. The evaluation is done using the validation dataset. Only the features that have a higher global relevance than any of the initial ones are selected.

For instance, suppose that the feature generation step in the current phase p produced a feature Z_i^p by combining two features from the previous phase named Z_j^{p-1} and Z_k^{p-1}. Then, the feature selection step would select Z_i^p if and only if

$$\mathrm{U}(Z_i^p, T^*; S'') > \max(\mathrm{U}(Z_j^{p-1}, T^*; S''), \mathrm{U}(Z_k^{p-1}, T^*; S'')) \tag{5.9}$$

5.3.5 Stopping criteria and other parameters

When two or more new features pass the selection criteria, the inductive process progresses to the next phase. Otherwise, the process stops and returns all new features selected during the execution.

Our implementation of the inductive approach also includes an optional stopping parameter that specifies the maximal number of phases. Many other parameters could be added to extend the process. For instance, the feature generation steps could consider all combinations of up to four features from the previous phase instead of two. Similarly, the selection process could be made indeterministic by randomly allowing some features to pass to the next phase even if they do not meet the criteria of Equation (5.9). Such parameters are worth considering since they help improve performance with other optimization processes (e.g., Neural Network, Evolutionary Strategies). However, in this research, we did not evaluate their potential effects with the proposed approach.

5.4 Application of the transformation models

The univariate and multivariate approaches generate a set of models $\Gamma_1, \Gamma_2, \ldots, \Gamma_r$ defining r new globally relevant features, where r is automatically determined. As shown in Figure 5.1,

[3]Overfitting is a general problem in data analysis that happens when the performance on the dataset used to generate the model is higher than the performance on an independent dataset.

the last step is to apply these models to add the new features to the test dataset or other independent datasets that need to be classified. Applying the models means computing either Equations (5.2) or (5.4) (for univariate or multivariate transformation, respectively) by replacing the X's and Y with the observed values and Ω with the parameter values for the given transformation model. Two issues may arise. The first one is the presence of missing values while the second issue happens when processing an instance with unseen value(s) for the explanatory attribute(s). The following two sections address these two issues. The third section explains the use of smoothing to improve the quality of features involving continuous explanatory attributes.

5.4.1 Missing values

We distinguish two situations depending on the type of the attribute for which we have missing values. In the first situation, one or more response attribute values are missing. In these cases, the transformation model dictates the addition of a constant to a missing value, which is not possible. Therefore, when the attribute Y takes a missing value, we simply return a missing value. In the second situation, one or several explanatory attributes have missing values but the response attribute value is not missing. Our implementation treats this situation explicitly by including the missing value as one of the potential values for all explanatory attributes. The program deals with the missing value just like the other values. For instance, when processing the dataset in Example 5, the program will create three subsets: S_1 for $X = a$, S_2 for $X = b$, and S_3 for $X =?$. For all non-empty subsets, the program creates a local relevance matrix and finds the corresponding parameter value. If there is no instance with $X =?$, then the corresponding ω value would be set to 0 (i.e., no transformation of Y when $X =?$). We named this transformation the *null* transformation. The null transformation is also used with multivariate transformations whenever one or several explanatory attribute values are missing.

5.4.2 Unseen values

The problem of unseen values arises when the model tries to process an instance for which the observed explanatory attribute value has not been seen during the generation of the model. Since the given value was not part of the training dataset, the models do not include an entry for this value and therefore there is no corresponding ω parameter. In this case, we also apply a null transformation. However, when several instances with the new value become available one may simply extend the models involving the given explanatory

attribute. This can be done in an incremental manner (i.e., without destroying the existing model) by using the algorithm `UnivBy2ThenReduceSearch` with two input relevance matrixes; the first one being the existing global relevance matrix and the second one the new local relevance matrix for the new value.

5.4.3 Smoothing the transformations

The transformations proposed in this research are step functions. This is appropriate when the explanatory attributes are nominal. However, when the explanatory attributes are numerical, the step function may not fit the initial relationship adequately. We address this problem during the application by smoothing the transformations for the models involving continuous explanatory attributes.

Figure 5.9 illustrates this problem. The figure at the top shows the initial relation between a response attribute y and an explanatory attribute x. Our approach recognized the negative effect of x on the global relevance of y and produced a new feature to reduce this effect. Figures (b) and (c) present this new feature against x and y, respectively, when smoothing is not used. These graphs clearly show that the initial relationship has been processed as a step function. The result is unsatisfactory; the pattern in the transformed space is not as smooth as the initial relationship and it is difficult to separate the positive and negative instances using the new feature. Moreover, we can easily identify some of the cut points that have been used to create the partition on x. Figures (d) and (e) show how the smoothing operation removes the roughness of the transformation. These two figures depict the graphs corresponding to (b) and (c), respectively, but when smoothing is used during the application of the transformation model. The resulting pattern is smoother and leads to a clear separation of the two classes.

We use the inverse distance weighting smoothing method [Isaaks & Srivastava, 1989] to modify the ω values in Equation (5.2) based on the observed values for the explanatory attribute. Let us suppose that $Z = \Gamma(X, Y; \Omega)$ and X is the discretized attribute from X^0, then

$$\text{val}_Z(s) = \text{val}_Y(s) + \omega_i^* \text{ if } \text{val}_X(s) = X(i) \tag{5.10}$$

where

$$\omega_i^* = \frac{\sum_{j=1}^{m} \omega_i d_j}{\sum_{j=1}^{m} d_j} \tag{5.11}$$

and

$$d_j = \frac{1}{\text{val}_{X^0}(s) - \overline{X}_j^0} \tag{5.12}$$

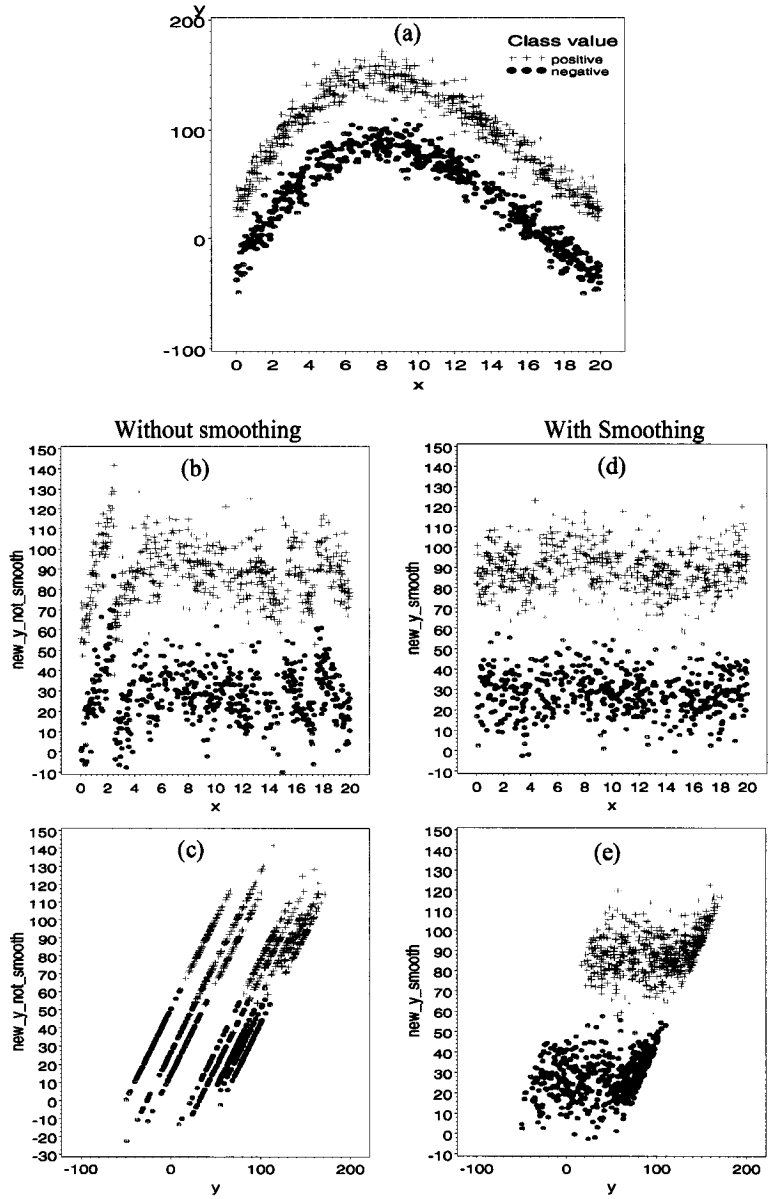


Figure 5.9: Example of the effect of smoothing. Figure (a) shows the initial relation between y and x. Figures (b) and (c) respectively present the new feature against x and y when smoothing is not used. Figures (e) and (d) respectively present the new feature against x and y when smoothing is used.

where $m = |\text{Dom}(X)|$, $\overline{X}_j^0$ is the average of X^0 in subset S_j, and $j = 1, 2, \ldots, m$. There are two extreme cases for which we set $\omega_i^* = \omega_i$. These are when $X^0 - \overline{X}_i^0 = 0$ or when $\text{val}_{X^0}(s)$ is either smaller (or greater) than the mininal (or maximal) value observed in the training set. In future work, we plan to experiment with other classes of smoothing function (e.g., splines and kriging).

5.5 Summary

This chapter completed the presentation of the GLOREF approach by explaining the generation of globally relevant features. The approach produces two types of features: the univariate and multivariate features. We first defined the types of transformations used to generate these features and the corresponding optimization problems. Then we presented the proposed solution for the univariate features. The presentation included the analysis of the problem space, the exhaustive solution, and various algorithms developed to ensure adequate performance. All algorithms presented use the relevance matrices obtained during the analysis of relevance (Chapter 4). Several examples have been introduced to illustrate the usefulness of the proposed algorithms.

We suggested two approaches to address the problem of automatic generation of multivariate features. The first one extends the univariate solution by applying a multivariate partitioning procedure. We explained the potential use and difficulties of this solution and then introduced a second approach. The latter approach follows an inductive process that considers multivariate transformations in progressive order of complexity. The algorithm exploits the univariate models already developed. This allows for a quick convergence during the optimization of the parameters. We discussed the approach we used to control the risk of overfitting as well as the stopping criteria.

Finally, we discussed the application of the transformation models. The issues to consider include the handling of missing and unseen values. We have also shown how smoothing is used to better fit the relationships observed between continuous attributes. The next chapter examines the feasibility of the GLOREF approach through an empirical evaluation of its performance on artificial and real world datasets. We will also discuss the limitations of the proposed solution.

Chapter 6

Empirical evaluation

6.1 Introduction

This chapter evaluates the feasibility of the GLOREF approach. There are three objectives. The first one is to assess the ability of the GLOREF approach to generate highly and globally relevant features. The second objective is to evaluate the usefulness of the new features with machine learning techniques. The last objective is to compare the GLOREF approach with alternative techniques.

The study includes both artificial and real world datasets. The artificial datasets are the ones used in Chapter 3 to study the potential merits of globally relevant features while the real world datasets are from the UCI repository [C. Blake & Merz, 1998]. This chapter is structured as follow. First, we describe the methodology in Section 6.2. Sections 6.3 and 6.4 discuss the results for the artificial and UCI datasets, respectively. Section 6.5 compares the GLOREF approach with the theoretical approach of Chapter 3 and two statistical techniques used to construct features in presence of attribute interactions. Section 6.7 elaborates on issues with the proposed study and on the limitations of the GLOREF approach. Finally, Section 6.8 summarizes this chapter.

6.2 Methodology

We assess the usefulness of the GLOREF approach by comparing the performance using the initial attributes with the performance using both the initial attributes and the new features. As in Chapter 3, we use the following notation to distinguish the initial attributes and the augmented feature sets:

$\mathcal{A}$: The initial set of attributes.

$\mathcal{F}$: The initial set of attributes augmented with the GLOREF features.

For completeness and consistency with Chapter 3, we also consider the effects of feature selection. The feature sets selected from the above sets of features are:

FS($\mathcal{A}$) : Feature subset selected from $\mathcal{A}$.

FS($\mathcal{F}$) : Feature subset selected from $\mathcal{F}$.

The primary criteria of performance is the accuracy of the model. We evaluate accuracy through a 10-fold cross-validation procedure. We repeat the procedure for all combinations of datasets and learning systems. In each fold, the procedure performs the following tasks:

Automatic discretization For each continuous attribute, the discretization step produces two new attributes using the *equal-width* and the *equal-frequency* intervals discretization methods (Section 4.2.3). The number of bins is set to 5 for the artificial dataset and to 3 for the UCI datasets[1]. The discretization methods determine the cut points using only the training data for the current fold.

Feature generation The GLOREF approach automatically generates new univariate and multivariate features using 70% of the training data available in current fold. The new features are added to the initial attributes to produce the augmented feature set $\mathcal{F}$.

Feature selection The feature selection produces the two feature subsets FS($\mathcal{A}$) and FS($\mathcal{F}$). This step uses the remaining 30% of the training data (i.e., the part that has not been used during feature generation, usually named the validation data).

Learning the models Using the given learning system and all training data for the current fold, we learn four models, varying the input attributes as follow:

1. All initial attributes ($\mathcal{A}$).
2. Subset of initial attributes (FS($\mathcal{A}$)).
3. All initial attributes and all GLOREF features ($\mathcal{F}$).
4. Subset of all initial attributes and GLOREF features (FS($\mathcal{F}$)).

Evaluation of the models We extend the test dataset for the current fold with the new features (discretized and GLOREF features) and we evaluate the accuracies of the four models generated above.

Upon completion of the 10^{th} fold, we obtain the final estimations of accuracy by averaging the results from the 10 folds. We obtain four accuracy estimations; one for each set

[1]It would have been probably preferable to also use 5 bins with most of the UCI datasets, but some of them were too small and we wished to keep the same parameters for all UCI datasets.

of models generated by the step 'Learning the models'. As in Chapter 3, we also report the standard deviation for each accuracy estimation.

We note that the GLOREF and feature selection tasks use different parts of the training sets. Without this precaution, the feature selection step may not be able to detect overfitting features. On the other hand, this split of the training data further reduces the data available to the GLOREF approach. Actually, only 63% of all the data is available to the GLOREF approach in any given fold (70% of the 90% for training equals $\approx$ 63%). In relatively small domains, this may limit the ability to handle complex relationships. The situation would have been even worse with a 5-fold procedure as suggested by other researchers [Dietterich, 1998] since only around 55% of all the data would have been available to the GLOREF approach in each fold.

We use the same feature selection method as in Chapter3 (Section 3.5) and 12 learning systems described in Appendix A with their default settings.

6.3 Artificial datasets

The artificial datasets are the ones introduced in Chapter 3, Section 3.4. The analysis of the results is divided in two parts. The first part discusses the global relevance of the new features while the second part considers the effects of the new features on the accuracy.

6.3.1 Analysis of the global relevance

The first part of the thesis statement requires the generation of globally relevant continuous features. To verify the achievement of this requirement, we compare the global relevance of the GLOREF features versus the global relevance of the initial attributes. We rely on two widely used measures of global relevance: the gain ratio and the χ^2 measures. We calculate the expected scores for these two measures for the best three initial attributes, the best three univariate GLOREF features, and the best three multivariate GLOREF features. Like for the estimation of the accuracy, we estimate the global relevance scores for the two measures by averaging the scores on test data obtained in the various folds of the cross-validation procedure.

Precisely, let t by the attribute type (initial, univariate GLOREF, or multivariate GLOREF) and A_j^i the attribute of type t with the i^{th} highest score on test data in fold j among all attributes of type t. The estimated score for the best i^{th} attribute of type t is given by

the following equation

$$S_t^i = 1/10 \sum_{j=1}^{10} score(A_j^i) \tag{6.1}$$

where *score* is the measure used (i.e., GainRatio or χ^2). For example, if $i = 1$, $t =$ 'initial', and *score* = GainRatio, then this statistic would report the expected gain ratio of the best initial attributes. Similarly, if $i = 2$, $t =$ 'GLOREF univariate', and *score* = GainRatio, then the statistic would report the expected gain-ratio of the second best univariate GLOREF feature. As mentioned above, we computed this statistic for the best three features of each type twice; once with the gain ratio measure and the other with the χ^2 measure.

Figures 6.1 and 6.2 present the results for the gain ratio and χ^2 measures, respectively. There is one chart for each artificial dataset. Each chart has three groups of three bars. The first group is for the best feature, the second group for the second best feature, and the third group displays the relevance of the third best feature. The three bars within each group represent the scores for the three feature types: initial, GLOREF univariate, and GLOREF multivariate, respectively.

We first notice the consistency between the two measures since the charts in Figures 6.1 and 6.2 are very similar. In particular, both measures conclude that the global relevances of the best GLOREF features are higher than the global relevance of the best initial attributes for all datasets. On the other hand, the importance of the increase in global relevance fluctuates considerably from one dataset to another. This variability was expected since the importance of attribute interactions varies greatly between the artificial datasets (Section 3.4). Table 6.1 details the increases in global relevance between the best initial attribute and the best GLOREF feature for each dataset.

The datasets with the largest increase in global relevance are: N4MN and N5MN. For N4MN, the gain ratio of the best initial attribute is .17 while the best GLOREF feature has a gain ratio of .98, for an increase of .81 (or 488%). For the dataset N5MN, we also obtained an increase in gain ratio of .81 from the best initial attribute (.19) to the best GLOREF feature (1.0). The χ^2 measure reports similar statistics for these two datasets. The following datasets in decreasing order of increased in gain ratio are: N4F1 (.53), N4F1 (.34), N1F1 (.34), N6F1 (.26), N6MN (.25), N3MN (.24), N2F1 (.20), N5F1 (.20), N2MN (.15), and N1MN (.14). All improvements are highly statistically significant since the p-values for both measures were all $< .0001$. These results combined with Figures 6.1 and 6.2 demonstrate that the GLOREF approach succeeded in producing new highly globally relevant features. We now consider the usefulness of the new features for learning.

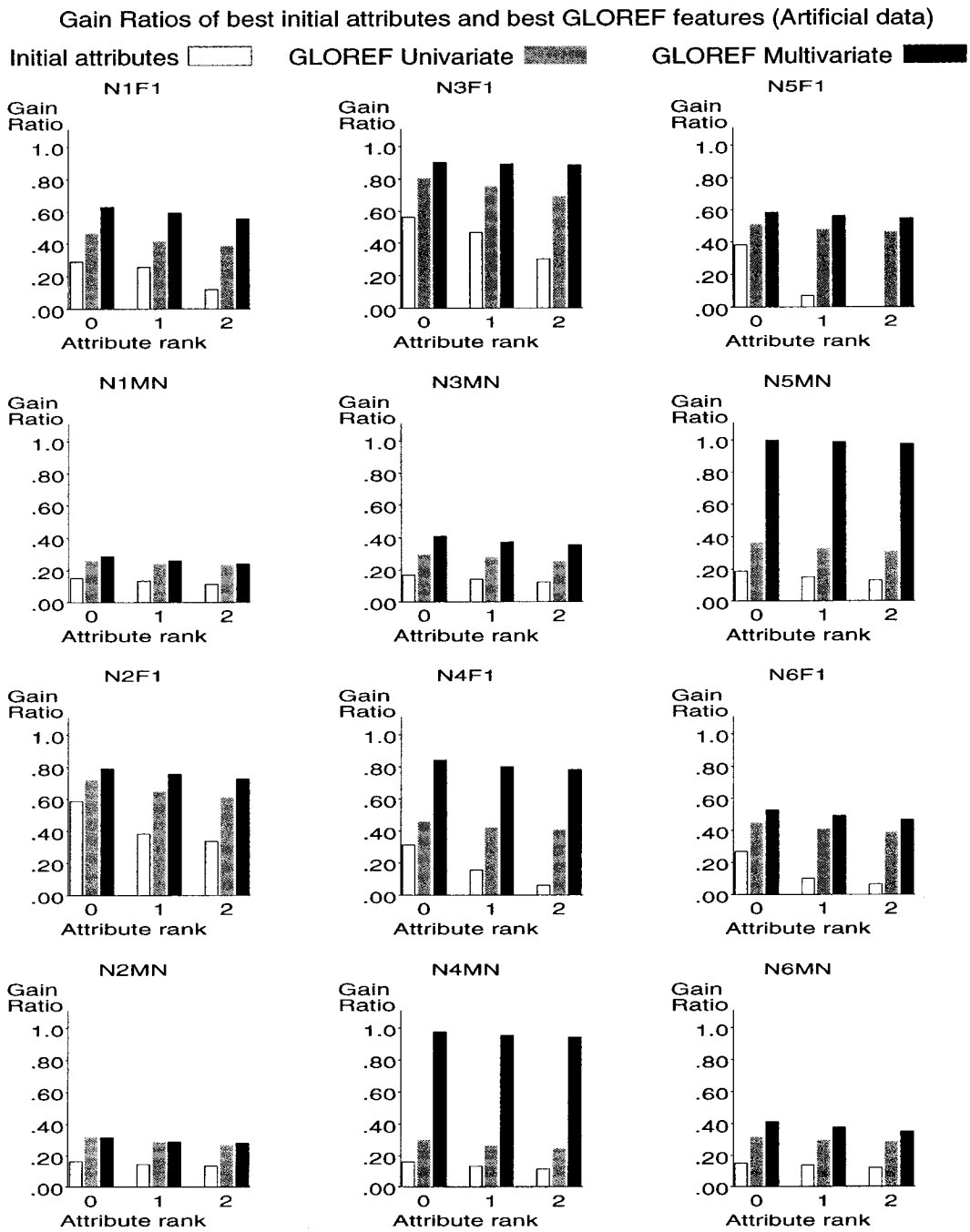


Figure 6.1: Global relevance based on gain ratio of the best 3 initial attributes, GLOREF univariate features, and GLOREF multivariate features for the 12 artificial datasets.

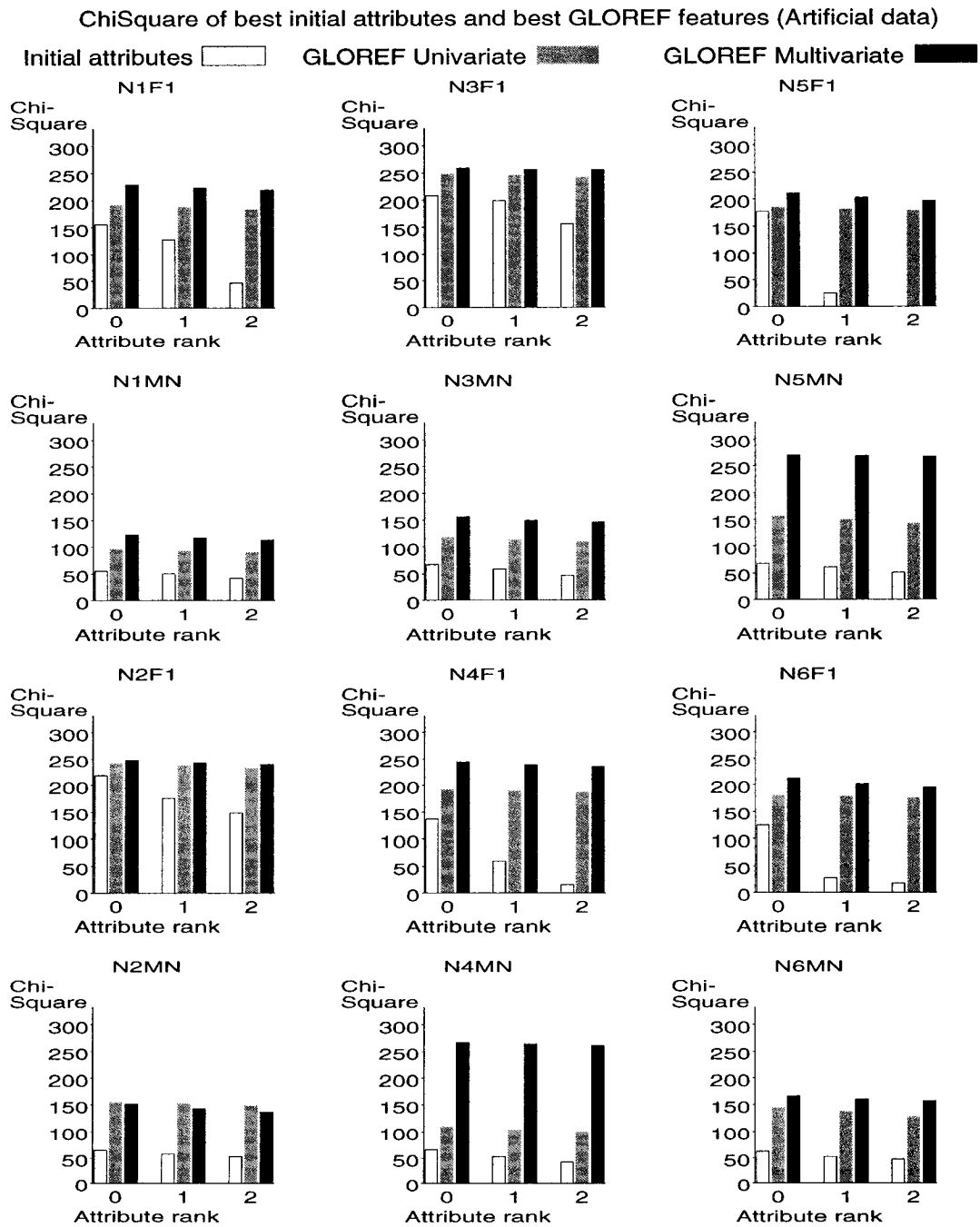


Figure 6.2: Global relevance based on the χ^2 measure of the best 3 initial attributes, GLOREF univariate features, and GLOREF multivariate features for the 12 artificial datasets.

Dataset	Best Initial		Best GLOREF					
			GainRatio			χ^2		
	GR	χ^2	Type	Score	Diff (%)	Type	Score	Diff (%)
N1F1	.29	156	Mul	.63	.34 (117 %)	Mul	229	73 (47 %)
N1MN	.15	55	Mul	.29	.14 (91 %)	Mul	123	67 (121 %)
N2F1	.59	219	Mul	.79	.20 (34 %)	Mul	248	29 (13 %)
N2MN	.17	66	Uni	.32	.15 (88 %)	Uni	155	89 (135 %)
N3F1	.56	208	Mul	.90	.34 (60 %)	Mul	259	51 (25 %)
N3MN	.17	66	Mul	.41	.24 (145 %)	Mul	156	90 (135 %)
N4F1	.31	137	Mul	.84	.53 (169 %)	Mul	245	108 (79 %)
N4MN	.17	67	Mul	.98	.81 (488 %)	Mul	266	200 (300 %)
N5F1	.38	177	Mul	.58	.20 (53 %)	Mul	211	34 (19 %)
N5MN	.19	68	Mul	1.0	.81 (435 %)	Mul	270	202 (298 %)
N6F1	.27	125	Mul	.53	.26 (97 %)	Mul	212	87 (69 %)
N6MN	.16	63	Mul	.41	.25 (163 %)	Mul	166	103 (164 %)

Table 6.1: Comparison of global relevance between the best initial attribute and the best Global GLOREF feature based on the gain ratio and the χ^2 measures for each artificial dataset.

6.3.2 Accuracies with GLOREF features vs initial attributes

The second part of the thesis statement states that using highly globally relevant features increases the accuracy of several learning systems. The previous section demonstrated the ability of the GLOREF approach to generate highly globally relevant features for the artificial datasets considered. In this section, we analyse the effects of the new features on accuracy. We present results for all artificial datasets and the 13 learning systems considered.

To evaluate the improvements due to the use of the GLOREF features, we compare the accuracies obtained when using only the initial attributes versus the accuracies obtained when including the new GLOREF features. As explained in Section 6.2, we also take into account the potential effects of feature selection by building two additional models: one based on a subset of the initial attributes and the other based on a subset of the augmented feature set which includes the GLOREF features. Therefore, for each combination of datasets and classifiers, we obtain 4 accuracy estimations: all initial attributes, a subset of the initial attributes, all available features (initial and GLOREF), and a subset of all available features.

Tables C.1 to C.12 in Appendix C.1 present these estimated accuracies along with their corresponding standard deviations. There is one table for each dataset. Each row details the results for a given learning system (also called inducer). There are nine columns, but three of them are not discussed in this section ('Theo. All', 'Theo. FS', and 'GLOREF vs Theo.'). These three columns will be discussed in Section 6.5.1 when comparing results between the theoretical features (defined in Chapter 3) and the GLOREF features. We now describe the six columns of interest for the current section.

The first column gives the name of the learning system considered. The second column ('Initial att. All') displays the average accuracy and standard deviations for models involving all initial attributes. The third column ('Initial att. FS') presents the results when applying feature selection to select a subset of the initial attributes prior to learning. The following two columns ('GLOREF All' and 'GLOREF FS') show the accuracies and standard deviations when the learning systems also use the GLOREF features, without and with feature selection, respectively. The column with the maximal accuracy between the second and fifth columns appears in bold. Finally, the eighth column ('GLOREF vs Init.') displays the average difference in accuracy between the best model using the GLOREF features and the best model using only the initial attributes. The best model with the initial attributes is the one with maximal accuracy between columns 'Initial att. All' and 'Initial att. FS'. Similarly, the best model with the GLOREF features is the one with maximal accuracy between columns 'GLOREF All' and 'GLOREF FS'. A positive value in column 'GLOREF vs Init.' means that the models using the GLOREF features perform better than the model based on the initial attributes only and vice versa. A star besides the difference denotes significance at the 0.05 level using the standard t-test to compare group means.

Figure 6.3 offers a quick view of all the results combined. There is one point for each combination of learning systems and datasets for a total of 156 points (12 datasets * 13 learning systems). The horizontal position represents the best accuracy between columns 'Initial att. All' and 'Initial att. FG' while the vertical position denotes the best accuracy between columns 'GLOREF All' and 'GLOREF FS' for the given dataset and classifier system. The points located above the diagonal line indicate that the GLOREF features have had a positive effect on accuracy. All points below the diagonal line represent experiments in which the use of the GLOREF features resulted in a decrease in accuracy.

Figure 6.3 clearly shows the strong positive effects of the GLOREF features on the accuracy. Moreover, the figure shows that extending the data representation with the GLOREF features rarely reduced the accuracy significantly for the artificial datasets. On the other hand, the importance of the positive effects of the new features is considerable

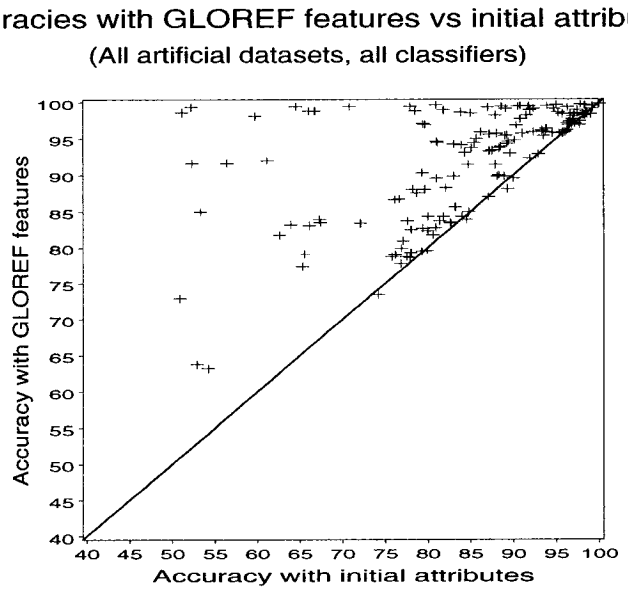


Figure 6.3: Accuracies using GLOREF features and initial attributes versus accuracies with initial attributes only.

since we find many points located far above the diagonal line.

Table 6.2 reports the gains in accuracy for all combinations of datasets and learning systems. There are two lines for each classifier. The first one shows the differences in accuracy between the best model with the initial attributes and the best model when the GLOREF features are used. The second line presents the percentage of increased (or decreased) in accuracy with respect to the best model using only the initial attributes. A star besides an accuracy difference indicates statistical significance at level 0.05. To analyze the results, we group them by dataset and then by learning system.

Table 6.3 summarizes the results by dataset. Each row accounts for the results from all learning systems with the dataset specified in the first column. The following four columns report statistics on the difference between the best GLOREF model and the best model with the initial attributes only. These statistics are computed from the values in column 'GLOREF vs Init' in Tables C.1 to C.12. The next three columns report the number of times that the GLOREF features increased accuracy, decreased accuracy, and did not change the accuracy, respectively. The sum of these three columns is 13 since we used 13 learning algorithms. The last two columns provide the number of times we obtained a statistically significantly better accuracy and a statistically significantly worse accuracy, respectively (at level 0.05).

Classifier	N1F1	N1MN	N2F1	N2MN	N3F1	N3MN	N4F1	N4MN	N5F1	N5MN	N6F1	N6MN
BaggingDT	0.40 0.4%	0.20 0.3%	0.50 0.5%	−1.2 −1%	0.20 0.2%	0.20 0.2%	0.00* 0.0%	4.40* 4.6%	4.70* 5.2%	7.70* 8.4%	6.40* 7.2%	3.50* 4.0%
BoostingDT	0.10 0.1%	1.20 1.5%	0.80 0.8%	−.50 −.6%	−.60 −.6%	0.90 1.1%	0.00* 0.0%	3.00* 3.1%	4.70* 5.2%	4.60* 4.8%	6.30* 7.1%	1.70 1.9%
DecisionStump	11.2* 13%	13.6* 21%	2.50 2.7%	16.1* 24%	6.70* 7.3%	11.3* 16%	17.3* 22%	32.8* 50%	0.90* 1.0%	28.6* 40%	8.50* 10%	16.6* 25%
DecisionTable	9.70* 11%	1.30 1.7%	0.50 0.5%	4.30* 5.4%	1.40 1.4%	6.10* 7.9%	14.8* 18%	20.3* 26%	5.40* 6.1%	21.5* 28%	13.6* 17%	8.80* 11%
HyperPipes	35.2* 62%	9.00* 17%	6.50 7.2%	10.9* 21%	10.3* 12%	22.0* 43%	38.4* 64%	47.1* 90%	30.9* 51%	47.5* 93%	39.3* 75%	31.7* 59%
IB1	0.10 0.1%	−.80 −1%	0.00 0.0%	0.70* 0.8%	−.10 −.1%	3.00 3.9%	2.70* 2.8%	1.90* 1.9%	8.80* 10%	8.70* 9.6%	8.60* 10%	10.5* 14%
IB5	−.10 −.1%	0.70 0.9%	−.30 −.3%	0.30* 0.4%	−.20 −.2%	2.30 2.8%	3.40* 3.6%	0.30* 0.3%	5.20* 5.8%	5.60* 6.0%	8.40* 9.6%	6.80* 8.2%
J48	3.60* 3.9%	2.90* 3.8%	0.10 0.1%	2.40* 2.9%	1.10 1.1%	4.50* 5.8%	6.80* 7.4%	12.3* 14%	5.10* 5.7%	10.8* 12%	5.90* 6.7%	8.40* 11%
KernelDensity	−.10 −.1%	1.00 1.3%	−.10 −.1%	−.60* −.7%	0.00 0.0%	3.80 4.9%	1.80* 1.9%	1.30* 1.3%	6.10* 7.0%	7.70* 8.4%	9.00* 11%	10.1* 13%
NaiveBayes	7.70* 8.8%	−.30 −.4%	−.80 −.8%	1.10* 1.4%	0.20 0.2%	1.90* 2.3%	13.5* 16%	17.1* 21%	3.40* 3.8%	18.7* 23%	10.2* 12%	9.80* 13%
OneR	13.3* 16%	12.1* 19%	2.80 3.0%	17.0* 26%	6.90 7.6%	19.1* 31%	17.6* 22%	32.1* 48%	1.70* 1.9%	34.9* 54%	11.0* 14%	19.3* 30%
PART	1.90 2.0%	2.80* 3.7%	0.80 0.8%	2.50* 3.1%	1.10 1.1%	3.30* 4.2%	7.00* 7.6%	10.0* 11%	5.40* 6.1%	8.90* 9.8%	9.30* 11%	6.10* 7.4%
SMO	0.60 0.6%	2.30 2.9%	0.80 0.8%	6.70* 7.9%	−.10 −.1%	−.10 −.1%	1.50* 1.5%	−.10 −.1%	1.30 1.4%	NA	2.20* 2.3%	0.40 0.4%

Table 6.2: Accuracy improvements by dataset and by classifier between models with the GLOREF features and models with the initial attributes only for the artificial datasets.

GLOREF vs initial attributes by dataset

Dataset	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
N1F1	6.43	9.89	-.10	35.2	11	2	0	6	0
N1MN	3.54	4.80	-.80	13.6	11	2	0	5	0
N2F1	1.08	1.92	-.80	6.50	9	3	1	0	0
N2MN	4.59	6.27	-1.2	17.0	10	3	0	10	1
N3F1	2.07	3.51	-.60	10.3	8	4	1	2	0
N3MN	6.02	7.11	-.10	22.0	12	1	0	7	0
N4F1	9.60	10.8	0.00	38.4	11	0	2	11	0
N4MN	14.0	15.1	-.10	47.1	12	1	0	12	0
N5F1	6.43	7.65	0.90	30.9	13	0	0	12	0
N5MN	15.8	13.9	-.20	47.5	12	1	0	12	1
N6F1	10.7	9.03	2.20	39.3	13	0	0	13	0
N6MN	10.3	8.33	0.40	31.7	13	0	0	11	0
All	7.54	9.75	-1.2	47.5	135	17	4	101	2

Table 6.3: Comparison, by dataset, of accuracies between models with the GLOREF features and models with the initial attributes only for the artificial datasets.

The average improvements vary from 1.08 (N2F1) to 15.8 (N5MN). As expected, the improvements tend to be more important in domains with stronger attributes interactions (i.e., the last 6 domains) than in domains with simpler interactions (i.e., the first 6 domains). As indicated by the second column, the standard deviation is fairly important in all domains. This variability signifies that the learning algorithms are not uniformly affected by the use of the GLOREF features. This was also expected since some of the learning algorithms used are more reliable than others with respect to attribute interactions.

From the column 'Min', we observe that the use of the GLOREF features never reduced the accuracy by more than 1.2. On the other hand, the column 'Max' indicates that the GLOREF features lead to increases in accuracy of up to 47.5. These results demonstrate that the risk that the GLOREF features reduce the accuracy is very low whereas the potential gains in accuracy are very important. The potential benefits of the GLOREF features are further illustrated by the last five columns in Table 6.3. Among the 156 cross-validation experiments performed, the GLOREF features increased the accuracy in 135

cases (≈ 87%), reduced the accuracy in 17 cases (≈10.9%), and had no effect on the accuracy in 4 experiments (≈2.5%). When considering only the statistically significant differences, we find that the GLOREF features improved the accuracy in 101 cases (≈65%) while reducing it in only 2 cases (≈1.2%). Such a high proportion of statistically significant positive results is very convincing. In particular, when we take into account the fact that the high standard deviations observed make it difficult to achieve significance.

It is revealing to link the improvements in accuracy with the increases in global relevance between the best GLOREF feature and best initial attribute (Table 6.1). We remark that the two datasets with the largest improvements in accuracy, N5MN (15.8) and N4F1 (14.0), correspond to the datasets for which we observed the largest improvements in global relevance. Inversely, the datasets with a lower improvement in global relevance tend to also have a more modest increase in accuracy. There are few exceptions such as domain N5F1 for which we obtained a fairly low increase in global relevance (.2 in gain ratio) but a respectable gain in accuracy (6.43). These results lead to the following conclusions:

1. A high improvement in global relevance between the best GLOREF feature and the best initial attribute leads to the an important gain in accuracy.

2. In some domains, relatively small improvements in global relevance also lead to important gain in accuracy.

Table 6.4 stratifies the results by learning systems. The format is the same as with Table 6.3. The HyperPipes system is the one that profited the most from the GLOREF features with an average accuracy improvement of 27.4. The following classifiers in decreasing order of average accuracy improvement are: OneR (15.7), DecisionStump (13.8), DecisionTable (8.98), NaiveBayes (6.88), J48 (5.33), PART (4.93), IB1 (3.68), KernelDensity (3.33), IB5 (2.70), BaggingDT (2.25), BoostingDT (1.85), and SMO (1.27). The standard deviations are relatively important which indicates that the extent of the improvements due to the use of the GLOREF features is not consistent over the various datasets for any given learning system. Nevertheless, all learning systems have been significantly positively affected by the GLOREF features since the number of significantly better accuracies vary from 3 out of 12 (25%) to 11 out of 12 (≈92%). The only two significantly negative results have been obtained with the KernelDensity and the SMO learning systems.

From these results, we conclude that all learning algorithms can be significantly positively affected by the GLOREF features. Moreover, the importance of improvements appears to be inversely proportional to the ability of the learning systems to handle attribute interactions. For instance, the performances of simple methods such as the HyperPipes and

GLOREF vs initial attributes by classifier

Classifier	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
BaggingDT	2.25	2.94	-1.2	7.70	10	1	1	5	0
BoostingDT	1.85	2.28	-.60	6.30	9	2	1	4	0
DecisionStump	13.8	9.49	0.90	32.8	12	0	0	11	0
DecisionTable	8.98	7.21	0.50	21.5	12	0	0	9	0
HyperPipes	27.4	15.1	6.50	47.5	12	0	0	11	0
IB1	3.68	4.22	-.80	10.5	9	2	1	7	0
IB5	2.70	3.10	-.30	8.40	9	3	0	7	0
J48	5.33	3.74	0.10	12.3	12	0	0	10	0
KernelDensity	3.33	3.90	-.60	10.1	8	3	1	6	1
NaiveBayes	6.88	6.97	-.80	18.7	10	2	0	9	0
OneR	15.7	10.2	1.70	34.9	12	0	0	10	0
PART	4.93	3.31	0.80	10.0	12	0	0	9	0
SMO	1.27	1.92	-.20	6.70	8	4	0	3	1
All	7.54	9.75	-1.2	47.5	135	17	4	101	2

Table 6.4: Comparison, by classifier, of accuracies between models with the GLOREF features and models with the initial attributes only for the artificial datasets.

OneR which are powerless with respect to attribute interactions increased more than the performances of sophisticated techniques such as decision trees (J48), instance-based (IB1 and IB5), and support-vector machines (SMO). This result is consistent with our introductory statement about the need for such an approach (Chapter 1) as well as the results of our study of the effects of attribute interactions (Chapter 3).

6.4 UCI datasets

To further evaluate the feasibility of the GLOREF approach, we experimented with the 12 UCI datasets described in Table 6.5. These are the first 12 datasets in alphabetical order from the repository that have at least one continuous attribute. As we did in the previous section, we shall first evaluate the global relevance of the new features generated by the GLOREF approach and then analyze the impacts of the new features on the accuracy.

Dataset	Numerical	Nominal	Class	Size	Default Acc
autos	15	11	7	205	67.3%
balance-scale	4	0	3	625	54.2%
breast-w	9	0	2	699	65.5%
cars	7	1	3	392	62.5%
colic	7	16	2	368	63.0%
credit-a	6	9	2	690	55.5%
diabetes	8	1	2	768	65.1%
glass	9	1	7	214	64.5%
heart-statlog	7	6	2	270	55.6%
hepatitis	2	18	2	155	79.4%
ionosphere	34	0	2	351	64.1%
liver-disorders	6	0	2	345	58.0%

Table 6.5: Summary of UCI datasets used.

6.4.1 Analysis of the global relevance

To evaluate the global relevance of the GLOREF features, we follow the approach introduced in Section 6.3.1. First, we summarize the global relevance of the best three GLOREF and initial features based on the gain ratio and χ^2 measures (Figures 6.4 and 6.5, respectively). There is one chart per dataset. Each chart presents the scores for the best three features of each type (i.e., initial, GLOREF univariate, and GLOREF multivariate). The scores are computed using Equation (6.1). Table 6.6 further details the increases in global relevance between the best initial attribute and the best GLOREF feature.

Based on the gain ratio measure, the GLOREF approach has been able to generate features with higher global relevance than the initial ones for all datasets. The χ^2 measure reports increased in global relevance for 11 domains out of 12. All the positive increases (for both measures) are statistically significant at .05 level while the only result with a lower global relevance is not significant.

As reported in Table 6.6, very large improvements in percentage are observed with the Liver and Balance-scale datasets with a respective increase in gain ratio (and χ^2) of 482% (475%) and 286% (519%). The increases in gain ratio for five of the remaining 10 datasets are between 60% and 70% while the improvements for the other five datasets are between 12% and 44%. The results with the χ^2 measure for these 10 datasets is slightly

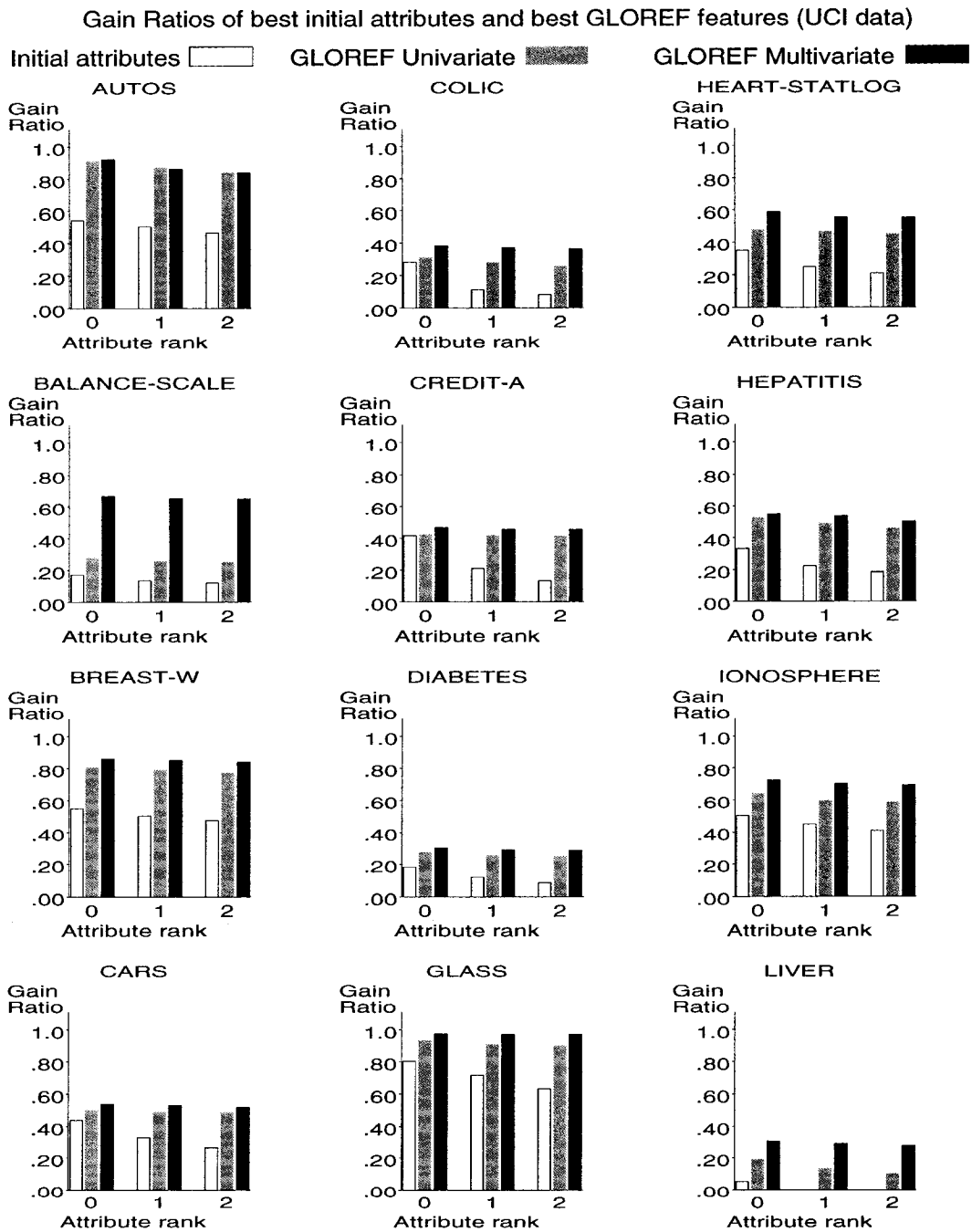


Figure 6.4: Global relevance of the best 3 initial attributes, GLOREF univariate features, and GLOREF multivariate features for the 12 UCI datasets. The measure used is the average gain ratio over the 10 folds on the test sets (Equation (6.1)).

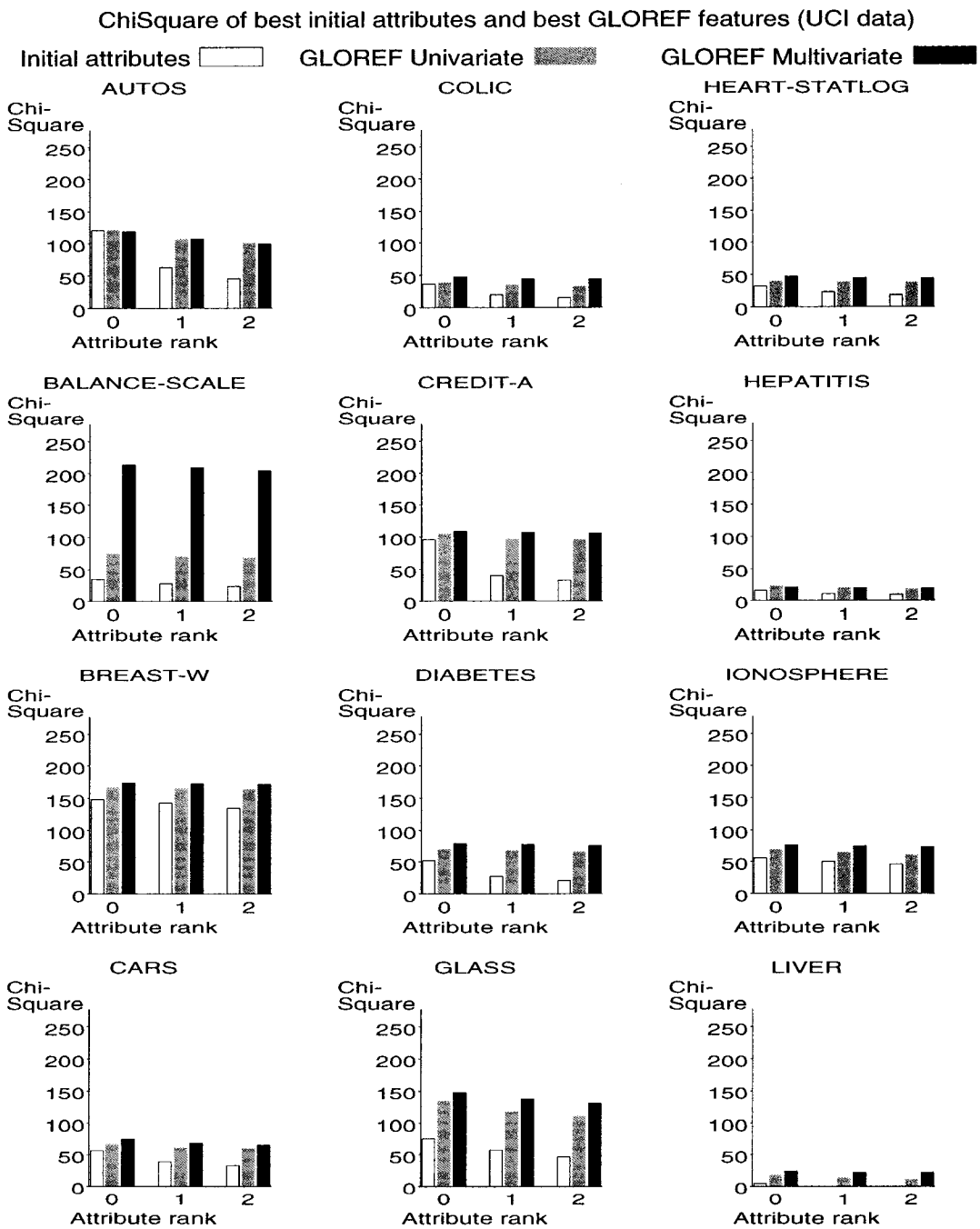


Figure 6.5: Global relevance of the best 3 initial attributes, GLOREF univariate features, and GLOREF multivariate features for the 12 UCI datasets. The measure used is the average gain ratio over the 10 folds on the test sets (Equation (6.1)).

Dataset	Best Initial		Best GLOREF					
			GainRatio			χ^2		
	GR	χ^2	Type	Score	Diff (%)	Type	Score	Diff (%)
autos	.55	121	Mul	.92	.37 (69 %)	Uni	121	-.1 (-.1 %)
balance-scale	.17	35	Mul	.67	.49 (286 %)	Mul	214	179 (519 %)
breast-w	.55	148	Mul	.86	.31 (57 %)	Mul	174	26 (18 %)
cars	.44	56	Mul	.54	.10 (23 %)	Mul	75	19 (33 %)
colic	.28	36	Mul	.38	.10 (36 %)	Mul	47	11 (31 %)
credit-a	.42	96	Mul	.47	.05 (12 %)	Mul	109	13 (14 %)
diabetes	.18	52	Mul	.30	.12 (64 %)	Mul	80	28 (53 %)
glass	.80	76	Mul	.98	.17 (21 %)	Mul	148	72 (96 %)
heart-statlog	.35	32	Mul	.59	.24 (68 %)	Mul	47	15 (47 %)
hepatitis	.33	15	Mul	.55	.22 (65 %)	Uni	22	7.3 (48 %)
ionosphere	.50	57	Mul	.73	.22 (44 %)	Mul	76	19 (34 %)
liver	.05	4.2	Mul	.31	.26 (482 %)	Mul	24	20 (475 %)

Table 6.6: Comparison of best initial and best Global GLOREF features based on gain ratio and the χ^2 measures for the UCI datasets.

different. We notice an increase in the global relevance of 96% for the Glass dataset followed by rises between 14% to 53% for 8 of the remaining ones, and finally, a decrease of .1% for the Autos dataset. As mentioned earlier, all positive results are statistically significant while the difference in χ^2 between the best initial attribute and the best GLOREF feature for the Autos dataset is not significant.

Although the two measures do not report identical improvements in global relevance, they are fairly consistent. This is verified by the similitude of patterns in corresponding charts from Figures 6.4 and 6.5. There are only 3 datasets out of 12 where the two measures appear to differ, these are: the Autos, the Hepatitis, and the Glass datasets. For the Autos and Hepatitis datasets, the gain ratio measure reports larger increases in global relevance than the χ^2 measure and vice versa for the Glass domain. In summary, the concordance between the two measures and the large increases observed in many domains make it clear that the GLOREF approach achieved the objective of producing highly globally relevant features. The next section considers the impact of these new globally relevant features on the accuracy.

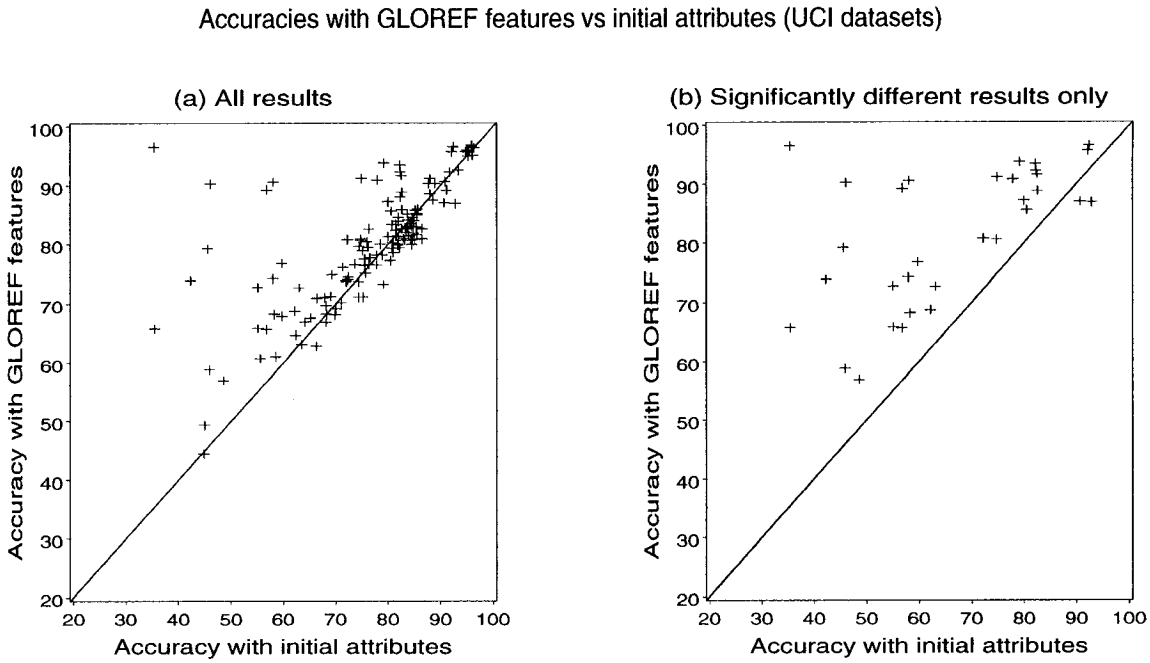


Figure 6.6: Figure (a) shows the accuracies using GLOREF features and initial attributes versus accuracies with initial attributes only for all experiments with UCI datasets. Figure (b) presents only the results where the difference in accuracy is significant.

6.4.2 Accuracies with GLOREF features vs initial attributes

For consistency, we shall discuss the effects of the new features on accuracy by following the same steps as in Section 6.3.2. That is, we first consider all results combined and then we specialize the analysis by dataset and by learning system.

The detailed results for the experiments involving the UCI datasets are provided in Tables C.13 to C.24 in Appendix C.1. These tables have six columns, which have been described in the beginning of Section 6.3.2. For convenience, we condense the average differences in accuracy for all experiments into Table 6.7. This table also provides the percentage of increase (or decrease) in accuracy between the best model using only the initial attributes and the best model using also the GLOREF features. As noted by the NAs in this table, there is no result with the SMO classifier for 6 datasets. This is because these datasets are multi-class problems and the implementation used could only handle binary learning tasks.

Classifier	AUTO	BALANCE	BREAST-W	CARS	COLIC	CREDIT-A	DIABETES	GLASS	HEART	HEPATITIS	IONOSPHERE	LIVER
BaggingDT	−1.8 −2%	11.2* 14%	0.00 0.0%	−3.8 −4%	−2.5 −3%	0.29 0.3%	0.92 1.2%	−3.2 −4%	2.59 3.2%	−3.8 −5%	0.58 0.6%	−.87 −1%
BoostingDT	−1.4 −2%	14.6* 18%	0.86 0.9%	−3.8 −4%	−.79 −1%	−.14 −.2%	1.70 2.4%	−4.1 −5%	0.37 0.5%	−2.7 −3%	−.85 −.9%	−1.7 −2%
DecisionStump	4.38 9.7%	32.5* 57%	4.15* 4.5%	4.57 6.9%	0.85 1.0%	−.29 −.3%	4.81 6.7%	−.45 −1%	5.19 7.0%	1.29 1.6%	5.70 6.9%	8.15 14%
DecisionTable	1.50 1.9%	16.3* 22%	0.44 0.5%	2.05 2.7%	−3.5 −4%	−.72 −.9%	−.53 −.7%	−3.5 −5%	−1.5 −2%	1.87 2.3%	−2.0 −2%	9.91* 17%
HyperPipes	17.7* 32%	44.2* 96%	61.2* 173%	9.67* 15%	31.7* 75%	17.1* 29%	30.3* 86%	12.9* 28%	33.7* 74%	−2.5 −3%	−6.0* −6%	6.55* 11%
IB1	3.40 4.5%	7.19* 9.0%	−.29 −.3%	6.10 8.2%	−1.9 −2%	0.00 0.0%	2.21 3.2%	2.01 2.8%	3.70 4.9%	1.92 2.4%	1.42 1.6%	2.29 3.7%
J48	−3.2 −4%	12.9* 17%	0.87 0.9%	−5.6 −6%	−4.4 −5%	−1.0 −1%	2.09 2.9%	−1.3 −2%	−.74 −.9%	3.25 3.9%	−.27 −.3%	2.37 3.6%
KernelDensity	−5.8 −7%	−.97 −1%	−1.0 −1%	5.83* 7.8%	−2.4 −3%	−1.4 −2%	−.91 −1%	5.63 8.1%	6.30 8.3%	−.67 −.8%	0.60 0.7%	2.84 4.4%
NaiveBayes	2.48 4.2%	−3.7* −4%	0.57 0.6%	1.54 2.3%	−2.5 −3%	5.07* 6.3%	0.25 0.3%	8.40* 17%	1.48 1.8%	−1.2 −1%	9.97* 12%	10.7* 19%
OneR	−.45 −.7%	32.5* 56%	3.58* 3.9%	−1.3 −2%	−.23 −.3%	−.58 −.7%	1.83 2.5%	5.15 9.3%	8.52* 12%	−.71 −.9%	6.29* 7.6%	8.86* 16%
PART	2.93 4.0%	9.14* 11%	0.87 0.9%	−2.0 −2%	−4.6 −5%	0.29 0.3%	−.65 −.9%	−.02 −.0%	4.44 5.9%	2.58 3.2%	3.14 3.6%	3.16 4.7%
SMO	NA	NA	NA	NA	−.80 −1%	0.58 0.7%	1.17 1.5%	NA	−3.0 −3%	NA	2.56 2.9%	16.3* 28%

Table 6.7: Accuracy improvements by dataset and by classifier between models with the GLOREF features and the models with the initial attributes only for the UCI datasets.

GLOREF vs initial attributes by dataset

Dataset	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
AUTO	1.80	6.13	-5.8	17.7	6	5	0	1	0
BALANCE-SCALE	16.0	14.7	-3.7	44.2	9	2	0	9	1
BREAST-W	6.48	18.2	-1.0	61.2	9	2	0	3	0
CARS	1.20	4.92	-5.6	9.67	6	5	0	2	0
COLIC	0.75	9.87	-4.6	31.7	2	10	0	1	0
CREDIT-A	1.59	5.15	-1.4	17.1	5	6	1	2	0
DIABETES	3.60	8.57	-.91	30.3	9	3	0	1	0
GLASS	1.95	5.48	-4.1	12.9	5	6	0	2	0
HEART	5.09	9.62	-3.0	33.7	9	3	0	2	0
HEPATITIS	-.06	2.38	-3.8	3.25	5	6	0	0	0
IONOSPHERE	1.77	4.19	-6.0	9.97	8	4	0	2	1
LIVER	5.71	5.28	-1.7	16.3	10	2	0	5	0
All	3.79	9.54	-6.0	61.2	83	54	1	30	2

Table 6.8: Comparison, by dataset, of accuracies between models with GLOREF features and models with initial attributes only for the real world datasets from UCI.

Figure 6.6 illustrates the results for the UCI datasets. The figure on the left side (Figure 6.6 (a)) includes all the results while the one on the right side (Figure 6.6 (b)) includes only the experiments with a statistically significant difference in accuracy between the best model using the GLOREF features and the best model using the initial attributes only. As explained in Section 6.3.2, the points located above the diagonal line indicate that the GLOREF features have had a positive effect on accuracy and vice versa.

Most of the points in Figure 6.6 (a) are located above the reference line but many of them appear very close to the line. Considering only the statistically significant results as shown in Figure 6.6 (b) clarifies the situation. From this figure, we recognize the benefits of the GLOREF features since there are many more points above the reference line than below it. Moreover, the only two points showing negative effects are much closer to the diagonal line than many of the points showing positive effects. This demonstrates that the overall positive effect largely surpasses the negative one.

Table 6.8 summarizes the results by dataset. The format of this table has been explained in Section 6.3.2. The improvement in accuracy varies from -0.6 (Hepatitis) to 16.0 (Balance-scale) with an overall average of 3.79. As expected, the standard deviations are relatively high which means that the actual improvements vary between the learning systems applied. For 11 out of 12 datasets, there is at least one experiment in which the use of the GLOREF features degraded the accuracy. In the worse case, the accuracy decreased by 6.0. On the other hand, we observed improvements in accuracy due to the use of the GLOREF features for all domains with maximal increased between 3.25 and 61.2.

From the 138[2] experiments performed, the GLOREF features increased the accuracy in 83 experiments (≈60%), decreased the accuracy in 54 experiments (≈39%), and did not affect the accuracy in 1 experiment (≈.7%). When considering only the statistically significant differences, we find that the GLOREF features improved the accuracy in 30 experiments (≈22%) while reducing it in only 2 experiments (≈1.4%). The high proportion of statistically significant positive results demonstrates the usefulness of the GLOREF approach in real-world applications of machine learning.

As observed with the artificial datasets, there is a link between the improvements in accuracy and the increases in global relevance between the best GLOREF feature and best initial attribute (Table 6.6). For instance, the two datasets with the highest increases in global relevance (i.e., the Balance-scale and Liver dataset) also lead to important increase in accuracy. Also consistent with earlier results, there are cases for which the improvements in global relevance appeared moderate while the gain in accuracy is important. The Breast-w dataset provides such an example with an improvement in global relevance of 57% and a gain in accuracy of 6.48.

Table 6.9 presents the results by classifier. As with the artificial datasets, the highest positive impacts has been observed with the HyperPipes classifier with an increase of 21.4. The following classifiers in decreasing order of average accuracy improvement are: DecisionStump (5.90), OneR (5.29), SMO (2.80), NaiveBayes (2.76), IB1 (2.34), DecisionTable (1.69), PART (1.61), KernelDensity (0.67), J48 (0.42), BoostingDT (0.16), and BaggingDT (-0.04). This order is similar to the ones obtained with the artificial datasets but there is one exception: the SMO classifier which went from the least influenced with the artificial datasets to the fourth most influenced with the UCI datasets. This might be related to the fact that the SMO classifier has been applied to only 6 datasets instead of 12 (i.e., only the two-class problems). Like before, the standard deviations are relatively im-

[2] 12 datasets * 12 learning systems - 6 multi-class datasets for which the SMO classifier did not work = 114 experiments.

GLOREF vs initial attributes by classifier

Classifier	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
BaggingDT	-.04	4.08	-3.8	11.2	6	6	0	1	0
BoostingDT	0.16	4.87	-4.1	14.6	4	8	0	1	0
DecisionStump	5.90	8.77	-.45	32.5	10	2	0	2	0
DecisionTable	1.69	5.81	-3.5	16.3	6	6	0	2	0
HyperPipes	21.4	19.6	-6.0	61.2	10	2	0	10	1
IB1	2.34	2.57	-1.9	7.19	9	2	1	1	0
J48	0.42	4.77	-5.6	12.9	5	7	0	1	0
KernelDensity	0.67	3.73	-5.8	6.30	5	7	0	1	0
NaiveBayes	2.76	4.78	-3.7	10.7	9	3	0	4	1
OneR	5.29	9.32	-1.3	32.5	7	5	0	5	0
PART	1.61	3.49	-4.6	9.14	8	4	0	1	0
SMO	2.80	6.85	-3.0	16.3	4	2	0	1	0
All	3.79	9.54	-6.0	61.2	83	54	1	30	2

Table 6.9: Comparison, by classifier, of accuracies between models with GLOREF features and models with initial attributes only for the real world datasets from UCI.

portant which indicates that the extent of the improvements varies over the various datasets. Concentrating on the significantly different results, we realize that all learning systems have been positively affected by the GLOREF features since the number of significantly better accuracies vary from 1 out of 12 (≈10%) to 10 out of 12 (≈83%). Only two significantly negative results have been obtained, one with the HyperPipes and one with the NaiveBayes.

In summary, the experiments with the UCI datasets confirmed the conclusions drawn from the experiments with the artificial datasets. In particular, the experiments shown that the GLOREF approach is capable of generating highly globally relevant features and the use of these features for learning often leads to significant increase of the accuracy. The gains in accuracy are generally higher with learning systems that strongly rely on the assumption of attribute independence but many significant gains have been obtained with more sophisticated systems. The extent of the improvements in accuracy tends to be proportional to the increase in global relevance between the best GLOREF feature and the best initial attribute. Finally, we noticed almost no negative results.

6.5 Comparisons with alternative methods

To complete this study of the feasibility of the GLOREF approach, we will now consider alternative techniques. We first review how the GLOREF features perform with respect to the theoretical features as derived in Chapter 3. Then, we contrast GLOREF to PCA (Principal Component Analysis) and ICA (Independent Component Analysis) which are two popular statistical techniques used to deal with non-independent attributes. Finally, we present a case study of the transformations generated by the GLOREF approach for one of the artificial domains and compare them to the transformations produced by PCA and ICA techniques.

We note that the theoretical features cannot be generated for the UCI datasets since they require knowledge that is not available. Therefore, we use only the artificial datasets to compare with the theoretical features. The PCA and ICA are restricted to continuous attributes. Among the UCI datasets used, only one has only continuous attributes (all others datasets have a mix of nominal and continuous attributes). It would be possible to use these techniques on the continuous attributes only and disregard the nominal attributes but this may bias the experiment in favor of the GLOREF approach which can use all types of attributes. To avoid such a bias, we also decided to use only the artificial datasets to perform the comparative study with the PCA and ICA methods.

6.5.1 GLOREF vs theoretical features

We introduced the theoretical features in Chapter 3 to study the potential improvements that could be achieved by exploiting information on attribute interactions. This study was idealized by assuming availability of perfect and complete knowledge of the relationships between the initial attributes. All theoretical features were derived from this perfect knowledge. The experiments performed showed that using the theoretical features augment the performances significantly in a variety of situations (i.e., with many learning systems and datasets). On the other hand, real world applications are rarely (if ever) accompanied by perfect knowledge about the attribute relationships. Therefore, we needed a technique that can extract the required information from the data itself. This is what the GLOREF method is doing. Even if they are not practical, the theoretical features provided an interesting starting point to evaluate realistic approaches such as the GLOREF framework. Hence, we now compare the results obtained with the GLOREF features with the results obtained with the theoretical features.

The experiment follows the methodology presented in Section 6.2, with modifications in the step 'Learning the models'. For the current experiment, this step reads:

Learning the models We run the given learning system for the current experiment four times, varying the input attributes as follow:

1. All initial attributes and all theoretical features.
2. Subset of initial attributes and theoretical features.
3. All initial attributes and all GLOREF features.
4. Subset of all initial attributes and GLOREF features.

All other aspects of the experiments are kept as described in Section 6.2.

The results for this section are provided in Tables C.1 to C.12 in Appendix C.1. The four columns of interest are: 'GLOREF All', 'GLOREF FS', 'Theo. All', 'Theo. FS', and 'GLOREF vs Theo.' The column 'GLOREF All' shows the accuracies and standard deviations when the learning systems use the initial and GLOREF features. Column 'GLOREF FS' provides the accuracies and standard deviations when feature selection is applied on the initial attributes and GLOREF features prior to learning. For the two columns involving the theoretical features ('Theo. All' and 'Theo. FS'), only the average accuracies are presented but the corresponding standard deviations can be found in Appendix B. Finally, the column 'GLOREF vs Theo.' indicates the difference in expected accuracy between models using the GLOREF features and models using the theoretical features. A positive value indicates that the GLOREF features lead to a higher accuracy than the theoretical features and vice versa. A star besides a difference indicates statistical significance at the 0.05 level.

Figure 6.7 displays the accuracies from the two sets of features for all experiments. As before, there is one point for each combination of learning system and dataset for a total of 156 points. The vertical axis stands for the expected accuracy using the GLOREF features while the horizontal axis represents the expected accuracy with the theoretical features. Tables 6.10 and 6.11 further summarize the results by dataset and by classifier, respectively. As shown by these tables, the GLOREF approach and the approach based on perfect knowledge of the dependencies are highly competitive with a slight advantage in favor of the theoretical approach. Considering all results combined, we note that the expected average accuracy using the theoretical features is higher than the expected average accuracy using the GLOREF features by 0.47. The GLOREF features produced a higher accuracy in 41 experiments out of the 156 ($\approx$26%), and 18 of these are statistically significant (11.5%). On

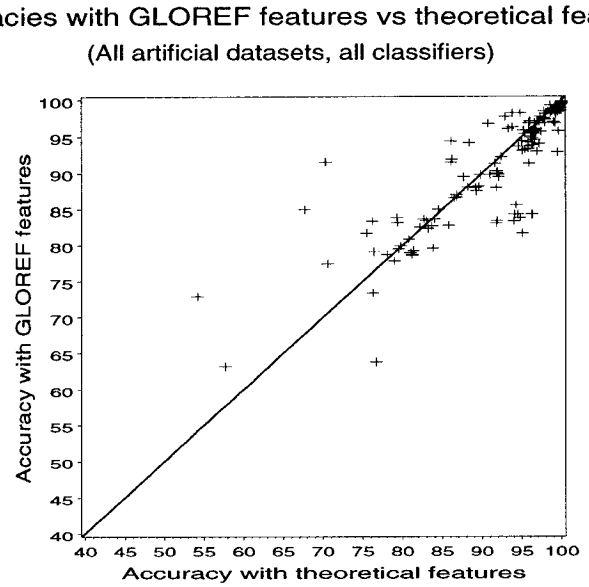


Figure 6.7: Accuracies using GLOREF features and initial attributes versus accuracies with theoretical features and initial attributes.

the other hand, there are 109 experiments (≈26%) for which we obtained a better accuracy with the theoretical features, 24 of these are statistically significant (≈15%).

The dataset N2MN is the only dataset for which the theoretical features significantly outperform the GLOREF features for all classifiers (Table 6.10). This dataset is also the only one for which the maximally relevant GLOREF feature is a univariate feature (Table 6.1). For all other artificial datasets, the best GLOREF feature comes from a multivariate transformation. The univariate GLOREF transformation discovered for this dataset is very useful since it greatly helps improving the performance over the initial attributes (Table 6.3), but not as good as the theoretical features. We remember that this dataset is somehow unique since all attributes are relevant (Table 3.1). With all other interaction structures, there is at least one of the influencing (or explanatory) attribute that is not relevant (i.e., not directly linked to the class attribute). It appears sensible that the GLOREF approach does not provide maximal benefits in domains where the initial attributes are already relevant, but more experiments would be required to further investigate this hypothesis.

As shown in Table 6.11, the classifiers OneR and HyperPipes tend to obtain better performance with the GLOREF features than with the theoretical features. The performance of these two classifiers is entirely determined by global univariate relevance. The fact that they perform so well with the GLOREF features clearly demonstrates the ability of the

GLOREF approach to combine complex information dispersed among several attributes into a single highly globally relevant feature. This unique ability comes from the inductive process developed to create the multivariate transformations. The theoretical features are not trying to combine more attributes than what is specified by the corresponding belief network and this may limit the total amount of relevant information in any given theoretical feature. On the other hand, the theoretical transformations optimize the independence which is particularly useful for an approach such as the NaiveBayes classifier. It is therefore not surprising to see that the performance of the NaiveBayes classifier is generally better with the theoretical features than with the GLOREF features.

In conclusion, the experiment described in this section shown that the GLOREF approach offers a performance that is competitive to an approach based on perfect knowledge of the attribute interactions across the domains considered. It also helped identify situations under which the GLOREF features are not as powerful as the theoretical features. In the following section, we focus on comparison between the GLOREF approach and two popular statistical techniques.

6.5.2 GLOREF vs PCA and ICA

This section considers two techniques often employed in the field of pattern recognition to pre-process data in which attribute interactions are expected. The first technique studied is the Principal Component Analysis (PCA), also named the Karhunen-Lòeve expansion. The second one is the Independent Component Analysis (ICA) method which recently emerged from research in Neural Networks. Both techniques have been described in Section 2.3.

The ICA and PCA techniques require continuous attributes only whereas the GLOREF approach is designed to use both continuous and nominal attributes. We did not want this fundamental difference to interfere in the evaluation process. Consequently, we decided to limit the experiment to the artificial datasets since they all contain continuous attributes only. To generate the PCA and ICA features, we used an optional package of the R system named `fastICA`. This package implements the FastICA algorithm [Hyvärinen et al., 2001]. The principal components are derived during a pre-whitening operation. In addition to the initial dataset, the function requires the user to specify the number of components to be extracted. We followed a common rule in practical applications of ICA by setting this number to the minimal number of principal components that would be required to account for more than 95% of the variability in the given dataset. We did not set any of the optional arguments except that we forced the algorithm to stop the optimization after 200 iterations regardless of the level of convergence achieved. This additional criteria did not affect our

GLOREF vs theoretical features by dataset

Dataset	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
N1F1	2.38	6.38	-1.7	21.3	4	7	2	3	0
N1MN	-.22	3.33	-4.2	6.80	4	8	1	2	0
N2F1	0.95	1.88	-1.0	6.00	9	4	0	3	0
N2MN	-8.3	4.68	-13	1.80	1	12	0	0	11
N3F1	0.80	2.00	-.90	4.80	6	6	1	3	1
N3MN	2.35	5.63	-3.0	18.7	10	3	0	3	0
N4F1	-1.7	1.75	-6.6	-.30	0	13	0	0	7
N4MN	-.35	0.42	-1.0	0.70	1	11	1	0	1
N5F1	-1.3	2.29	-3.9	5.80	1	12	0	1	2
N5MN	-.48	0.26	-1.3	-.20	0	13	0	0	2
N6F1	-.95	2.10	-3.1	5.50	1	12	0	1	1
N6MN	1.23	5.31	-3.8	17.3	4	8	1	2	1
All	-.47	4.38	-13	21.3	41	109	6	18	26

Table 6.10: Comparison, by dataset, of accuracies between models with GLOREF features and models with theoretical features for the artificial datasets.

experiments since the procedure converged with all datasets in less than 200 iterations.

We followed the methodology described in Section 6.2 with two simple modifications. First, we also ran PCA and ICA in the step named 'feature generation' to produce the the corresponding features. Second, we added the following feature sets to the step named 'Learning the models':

1. All initial attributes and all PCA features.
2. Subset of all initial attributes and PCA features.
3. All initial attributes and all ICA features.
4. Subset of all initial attributes and ICA features.

All other aspects of the methodology are kept as described in Section 6.2. The detail results for these sets of features are provided in Tables C.25 to C.36. The format of these tables is similar to the other tables in Appendix B, described in Section 6.3.2.

GLOREF vs theoretical features by classifier

Dataset	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
BaggingDT	-.95	1.87	-6.6	0.50	1	9	2	0	1
BoostingDT	-.43	1.32	-3.8	1.80	3	9	0	0	1
DecisionStump	0.93	4.45	-8.4	7.20	6	6	0	5	2
DecisionTable	-1.4	3.04	-10	1.00	4	8	0	0	2
HyperPipes	5.73	9.63	-13	21.3	9	3	0	8	3
IB1	-1.5	3.00	-11	0.20	3	9	0	0	2
IB5	-1.8	3.30	-12	0.00	0	11	1	0	2
J48	-1.4	2.49	-8.7	0.30	3	8	1	0	1
KernelDensity	-1.5	3.03	-11	0.10	2	10	0	0	1
NaiveBayes	-3.1	3.50	-13	-.40	0	12	0	0	7
OneR	1.51	4.84	-8.7	8.30	6	5	1	5	2
PART	-1.5	2.77	-9.8	1.10	2	9	1	0	1
SMO	-.66	1.27	-4.5	0.40	2	10	0	0	1
All	-.47	4.38	-13	21.3	41	109	6	18	26

Table 6.11: Comparison, by dataset, of accuracies between models with GLOREF features and models with theoretical features for the artificial datasets.

Figure 6.8 summarizes the usefulness of the PCA and ICA features on the accuracy. The figure on the left displays the accuracy for the models involving the PCA features versus the accuracy for the corresponding models built from the initial attributes only. The figure on the right presents the same information but for the ICA features versus the initial attributes. Both approaches generally improve the accuracies but the gains with the ICA features are more important than with the PCA features.

Figure 6.9 contrasts the usefulness of the GLOREF features versus the PCA and ICA features. As shown by Figure 6.9 (a), the GLOREF approach is clearly superior to the PCA technique. However, the ICA technique appears to be more competitive since most of the points are located near the diagonal reference line. To help compare the results between the GLOREF and ICA features, we provide a summary by dataset in Table 6.12. As shown by the last line in this table, the average accuracy with the GLOREF features is 1.64 higher than the average accuracy with the ICA features. The largest differences in accuracy

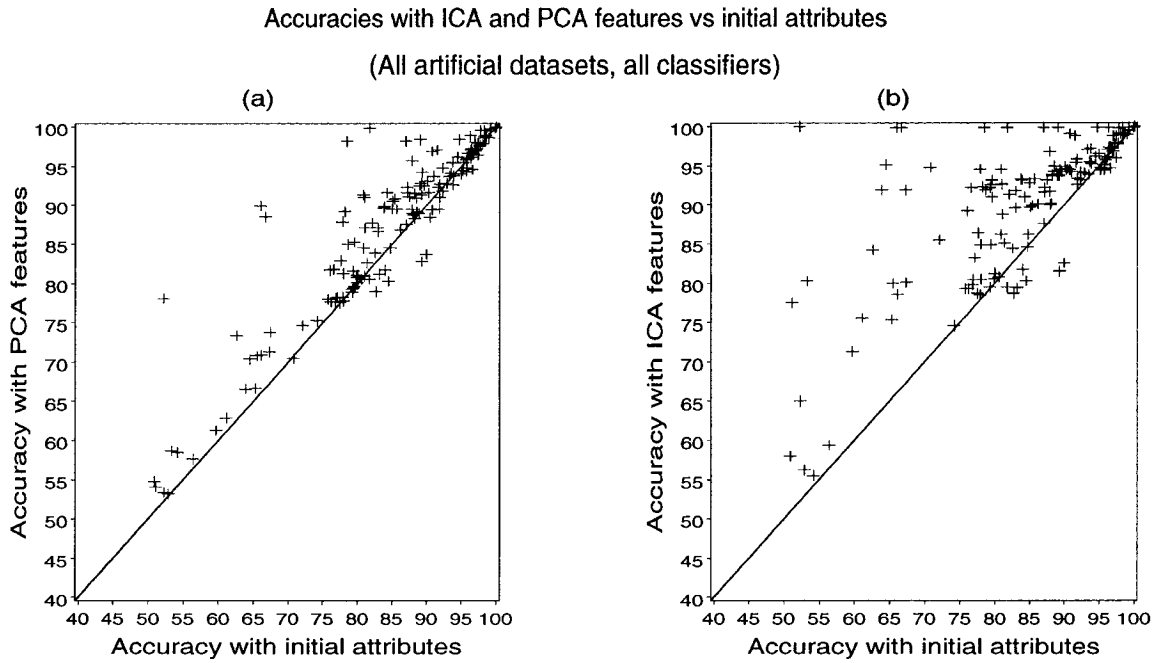


Figure 6.8: Figure (a) shows the accuracies using PCA features and initial attributes versus the accuracies with initial attributes only. Figure (b) shows accuracies using ICA features and initial attributes versus accuracies with initial attributes only. Results are for all artificial datasets and all classifiers.

improvement vary from 8.7 in favor of the ICA technique to 32.2 in favor of the GLOREF approach. Out of the 156 experiments, the GLOREF features produced a higher accuracy in 74 cases (≈47.4%), the ICA features lead to a higher accuracy in 75 cases (≈48%), and we observed no difference in the remaining 7 cases. The clearest distinction between the two techniques comes from the analysis of significance which favors the GLOREF approach with the 48 significantly better cases versus 18 significantly better cases for the ICA technique. It is interesting to note that the GLOREF approach is systematically better on some datasets such as N4F1 and N5MN. We do not have any explanation for the relatively low performance of ICA on these datasets.

In conclusion, the results of our experiments suggest that the GLOREF approach can generate features that are more adequate for machine learning than two commonly used statistical techniques. Moreover, it is important to remember that the GLOREF approach is also adequate in applications with a mixture of nominal and continuous attributes while the PCA and ICA can only handle continuous attributes. On the other hand, we recognize that

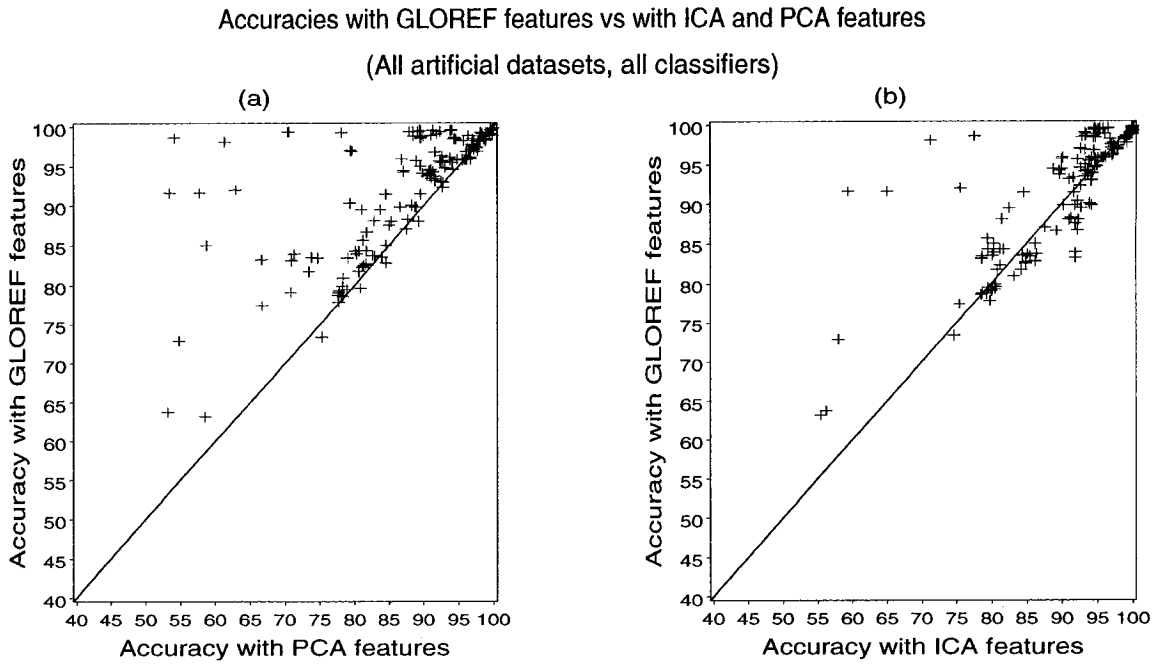


Figure 6.9: Figure (a) shows the accuracies using GLOREF features versus the accuracies with PCA features. Figure (b) shows accuracies using GLOREF features versus accuracies with ICA features. Results are for all artificial datasets and all classifiers.

the 12 artificial datasets considered may not be representative of a significant proportion of the real world datasets. Consequently, we think that more experimentations with real world datasets are required to better characterize the performance of the GLOREF approach with respect to the PCA and ICA techniques.

6.6 A case study

We now turn our attention to a specific case study to further illustrate the behavior of the GLOREF approach. The selected case is based on the N5MN dataset. This is the dataset for which the GLOREF technique realizes its maximal improvement in accuracy. This high improvement in accuracy is directly related to an automatically derived transformation that involves attributes X_1, X_2, and X_7. As shown in Table 3.1 (see the belief network named N5), the N5MN dataset has been designed such that X_1 and X_2 jointly reduce the relevance of X_7. It this section, we will review how the GLOREF technique automatically discovered this relationship and constructed a transformation that cancels its negative effects. We use

GLOREF vs ICA features by dataset

Dataset	Accuracy improvement				Better	Worse	Equal	Significantly	
	Avg	Std	Min	Max				Better	Worse
N1F1	4.58	8.61	0.00	32.2	11	0	2	5	0
N1MN	0.28	2.46	-2.0	7.70	4	7	2	1	0
N2F1	-.17	0.87	-1.0	2.30	3	9	1	1	0
N2MN	4.75	2.01	0.90	7.60	13	0	0	11	0
N3F1	-.44	0.63	-1.4	1.40	1	12	0	0	2
N3MN	-.57	4.73	-3.4	14.9	1	12	0	1	0
N4F1	5.55	6.68	-1.2	26.8	12	1	0	11	0
N4MN	-.67	0.36	-1.1	-.10	0	13	0	0	8
N5F1	0.92	5.04	-4.4	16.4	5	6	2	1	2
N5MN	5.29	4.93	-.20	21.0	12	1	0	12	0
N6F1	3.22	7.35	-3.2	26.6	11	2	0	4	0
N6MN	-2.9	3.52	-8.7	4.70	1	12	0	1	6
All	1.65	5.33	-8.7	32.2	74	75	7	48	18

Table 6.12: Comparison, by dataset, of accuracies between models with GLOREF features and models with ICA features for all artificial datasets.

relevance graphs and scatter plots to help illustrate the progression towards the discovered solution. We need to consider three transformations; two univariate transformations and one multivariate transformation. Finally, we will apply the learned multivariate transformation model to an independent test dataset and contrast the results with the best ICA and PCA features available for the same data.

For the example in this section, we use 70% (700 instances) of the initial N5MN dataset to learn the transformation models. For evaluation, we apply the learned models to the remaining 30% (or 300 instances). All the graphs shown to illustrate the usefulness of the transformations rely on the evaluation dataset only. As explained in Section 4.2.3, the analysis of relevance needs to discretize the numerical explanatory attributes X_1 and X_2 in order to analyse their effects on the relevance of X_7. In the relevance graphs and equations, we refer to the discretized version of X_1 and X_2 as $X1\_discretized$ and $X2\_discretized$, respectively. However, in the main text, we simplified the presentation by using only X_1 and X_2, whether we are referring to the discretized version or the initial one. We used the same width interval discretization technique with five bins (Section 4.2.3).

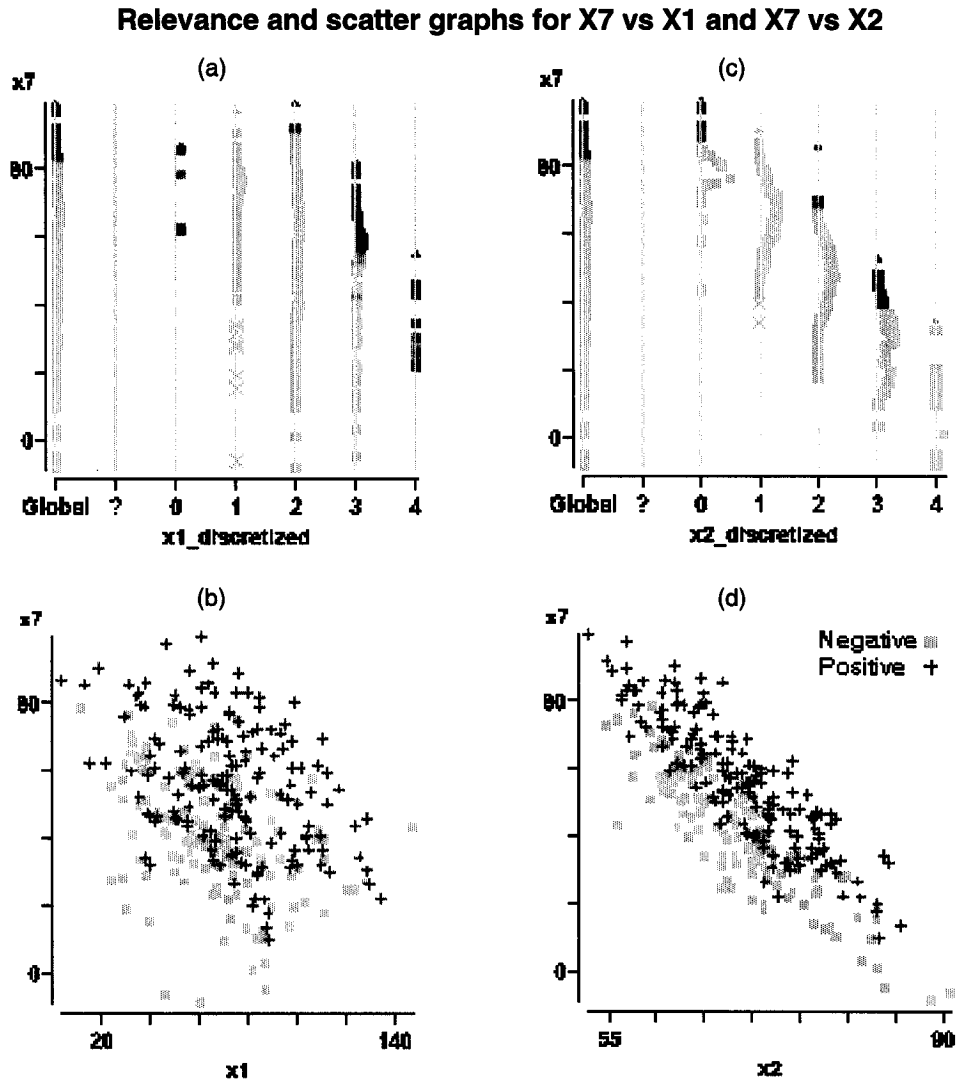


Figure 6.10: Relevance graphs and scatter plots showing the relations between the response attribute X_7 and the two explanatory attributes X_1 and X_2 in domain N5MN.

The initial situation is displayed in Figure 6.10. The two relevance graphs at the top show the effects of X_1 and X_2 on the relevance of X_7. We can see that both X_1 and X_2 negatively affect the relevance of X_7 by causing a misalignment of the most relevant cut points (Section 4.3.3). The relevance graph for X_2 is particularly convincing since it shows a high potential for a transformation of X_7. The two scatter plots at the bottom illustrate the lack of separability in the initial attribute space.

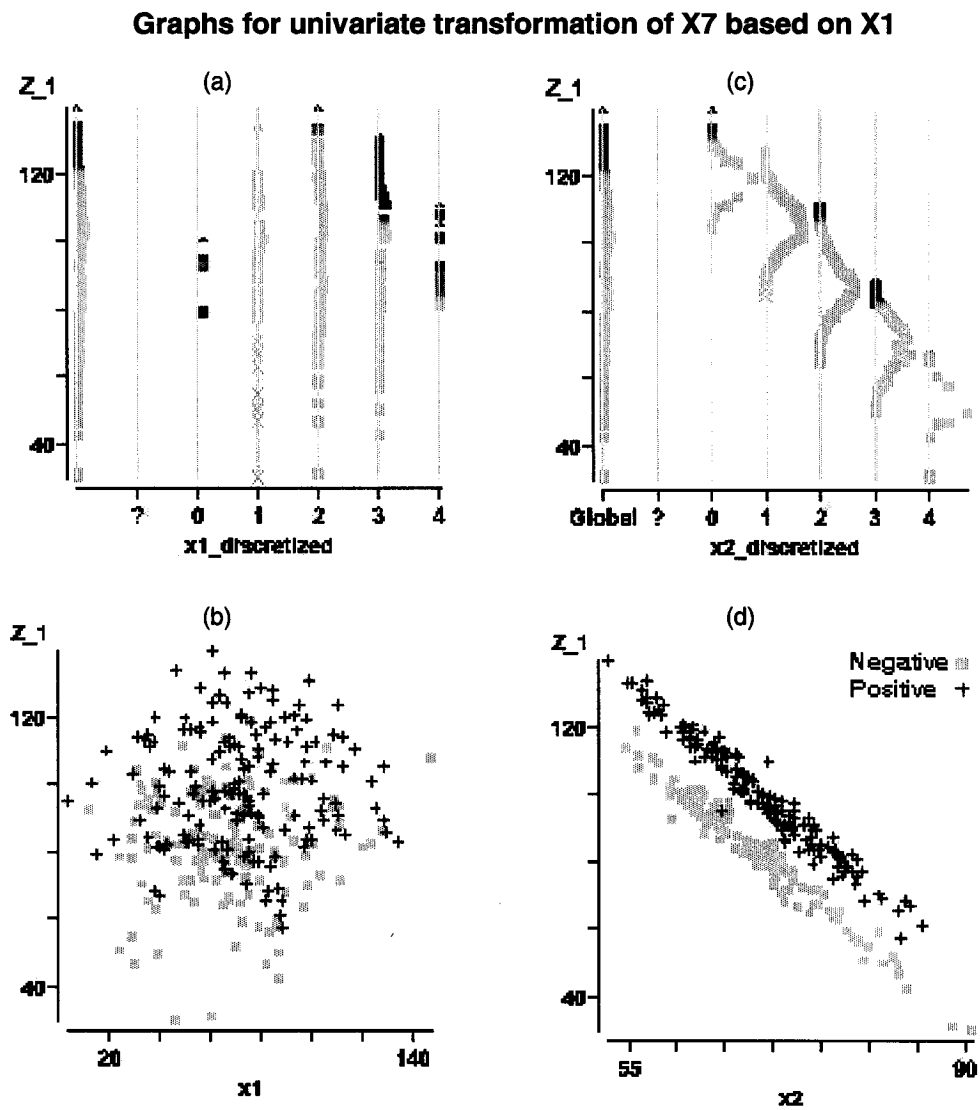


Figure 6.11: Relevance graphs and scatter plots showing the relations between the univariate GLOREF feature Z_1 and the two explanatory attributes X_1 and X_2 in domain N5MN.

The search for univariate transformations automatically generates a new features Z for each pair of attributes (Section 5.2). Adopting the notation of Chapter 5, we use $Z_1 = \Gamma(X1\_discretized, X_7; \Omega_1)$ to represent the transformation based on X_1 and X_7, and we use $Z_2 = \Gamma(X2\_discretized, X_7; \Omega_2)$ for the transformation based on X_2 and X_7, where Ω_1 and Ω_2 are the sets of parameters optimized by the search procedure. For the example

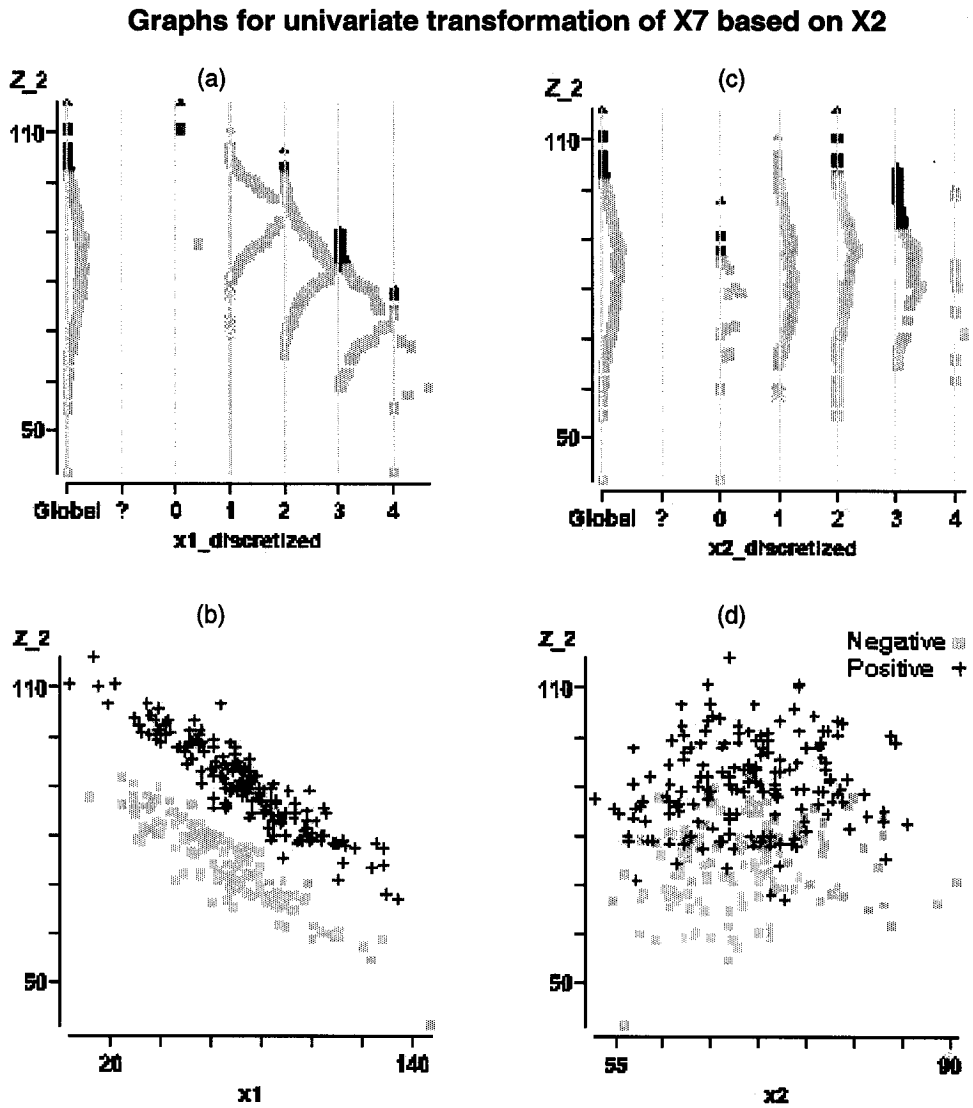


Figure 6.12: Relevance graphs and scatter plots showing the relations between the univariate GLOREF feature Z_2 and the two explanatory attributes X_1 and X_2 in domain N5MN.

considered, the solutions found by the GLOREF univariate search procedure are

$$\Omega_1 = \{0.0, 12.46, 34.81, 41.50, 49.63, 58.38\}$$

and

$$\Omega_2 = \{0.0, -1.25, 18.11, 38.14, 54.83, 66.03\}$$

Substituting in Equation (5.2), we have

$$Z_1 = \Gamma(X1\_discretized, X_7; \Omega = \{0.0, 12.46, 34.81, 41.50, 49.63, 58.38\})$$

$$= \begin{cases} X_7 + 0.0 & \text{if } X1\_discretized = ? \\ X_7 + 12.46 & \text{if } X1\_discretized = 0 \\ X_7 + 34.81 & \text{if } X1\_discretized = 1 \\ X_7 + 41.50 & \text{if } X1\_discretized = 2 \\ X_7 + 49.63 & \text{if } X1\_discretized = 3 \\ X_7 + 58.38 & \text{if } X1\_discretized = 4 \end{cases}$$

and

$$Z_2 = \Gamma(X2\_discretized, X_7; \Omega = \{0.0, -1.25, 18.11, 38.14, 54.83, 66.03\})$$

$$= \begin{cases} X_7 + 0.0 & \text{if } X2\_discretized = ? \\ X_7 + -1.25 & \text{if } X2\_discretized = 0 \\ X_7 + 18.11 & \text{if } X2\_discretized = 1 \\ X_7 + 38.14 & \text{if } X2\_discretized = 2 \\ X_7 + 54.83 & \text{if } X2\_discretized = 3 \\ X_7 + 66.03 & \text{if } X2\_discretized = 4 \end{cases}$$

Since the initial explanatory attributes (X_1 and X_2) are continuous, we applied a smoothing transformation while computing the values for Z_1 and Z_2 (Section 5.4.3). The usefulness of the new features Z_1 and Z_2 can be assessed through Figures 6.11 and 6.12, respectively. Figure 6.11 reveals that Z_1 strongly interacts with X_2 and also discloses the discriminative power of the data. Symmetrically, Figure 6.12 reveals a strong relationship with X_1 as well as the discriminative power of the data.

In the first phase of the search for multivariate relationships (Section 5.3), the GLOREF approach combines all pairs of univariate transformations. This leads to a new multivariate feature $Z_3 = \Gamma(\{X1\_discretized, X2\_discretized\}, X_7; \Omega_3)$. The exact definition found by the system for the example considered is:

$$Z_3 = \Gamma(\{X1\_discretized, X2\_discretized\}, X_7; \Omega_3)$$

$$= \begin{cases} X_7 + 0.00 & \text{if } X1\_discretized = ? \text{ or } X1\_discretized = ? \\ X_7 + 17.89 & \text{if } X1\_discretized = 0 \text{ and } X2\_discretized = 0 \\ X_7 + 31.87 & \text{if } X1\_discretized = 0 \text{ and } X2\_discretized = 1 \\ X_7 + 48.76 & \text{if } X1\_discretized = 0 \text{ and } X2\_discretized = 2 \\ X_7 + 68.25 & \text{if } X1\_discretized = 0 \text{ and } X2\_discretized = 3 \\ X_7 + 79.74 & \text{if } X1\_discretized = 0 \text{ and } X2\_discretized = 4 \\ X_7 + 27.30 & \text{if } X1\_discretized = 1 \text{ and } X2\_discretized = 0 \\ X_7 + 41.28 & \text{if } X1\_discretized = 1 \text{ and } X2\_discretized = 1 \\ X_7 + 58.17 & \text{if } X1\_discretized = 1 \text{ and } X2\_discretized = 2 \\ X_7 + 77.66 & \text{if } X1\_discretized = 1 \text{ and } X2\_discretized = 3 \\ X_7 + 89.15 & \text{if } X1\_discretized = 1 \text{ and } X2\_discretized = 4 \\ X_7 + 31.92 & \text{if } X1\_discretized = 2 \text{ and } X2\_discretized = 0 \\ X_7 + 45.90 & \text{if } X1\_discretized = 2 \text{ and } X2\_discretized = 1 \\ X_7 + 62.79 & \text{if } X1\_discretized = 2 \text{ and } X2\_discretized = 2 \\ X_7 + 82.28 & \text{if } X1\_discretized = 2 \text{ and } X2\_discretized = 3 \\ X_7 + 93.77 & \text{if } X1\_discretized = 2 \text{ and } X2\_discretized = 4 \\ X_7 + 43.08 & \text{if } X1\_discretized = 3 \text{ and } X2\_discretized = 0 \\ X_7 + 57.06 & \text{if } X1\_discretized = 3 \text{ and } X2\_discretized = 1 \\ X_7 + 73.95 & \text{if } X1\_discretized = 3 \text{ and } X2\_discretized = 2 \\ X_7 + 93.44 & \text{if } X1\_discretized = 3 \text{ and } X2\_discretized = 3 \\ X_7 + 104.93 & \text{if } X1\_discretized = 3 \text{ and } X2\_discretized = 4 \\ X_7 + 54.65 & \text{if } X1\_discretized = 4 \text{ and } X2\_discretized = 0 \\ X_7 + 68.63 & \text{if } X1\_discretized = 4 \text{ and } X2\_discretized = 1 \\ X_7 + 85.52 & \text{if } X1\_discretized = 4 \text{ and } X2\_discretized = 2 \\ X_7 + 105.01 & \text{if } X1\_discretized = 4 \text{ and } X2\_discretized = 3 \\ X_7 + 116.50 & \text{if } X1\_discretized = 4 \text{ and } X2\_discretized = 4 \end{cases}$$

Figure 6.13 displays the smoothed values obtained from this transformation. Clearly, the new feature Z_3 has a very high discriminating power and the initial X_1 and X_2 attributes are no longer affecting the relevance of Z_3.

To complete the example, we computed the principal components as well as the independent components from the same training set of 700 instances. Then, we applied all the transformation models (GLOREF, ICA, and PCA) to an additional dataset with 1000 novel instances from the same distribution. Figure 6.14 shows the best principal component obtained, the best independent component obtained, and the best GLOREF feature obtained (Z_3) versus the initial attributes X_1 and X_2. It is clear that all methods reduced

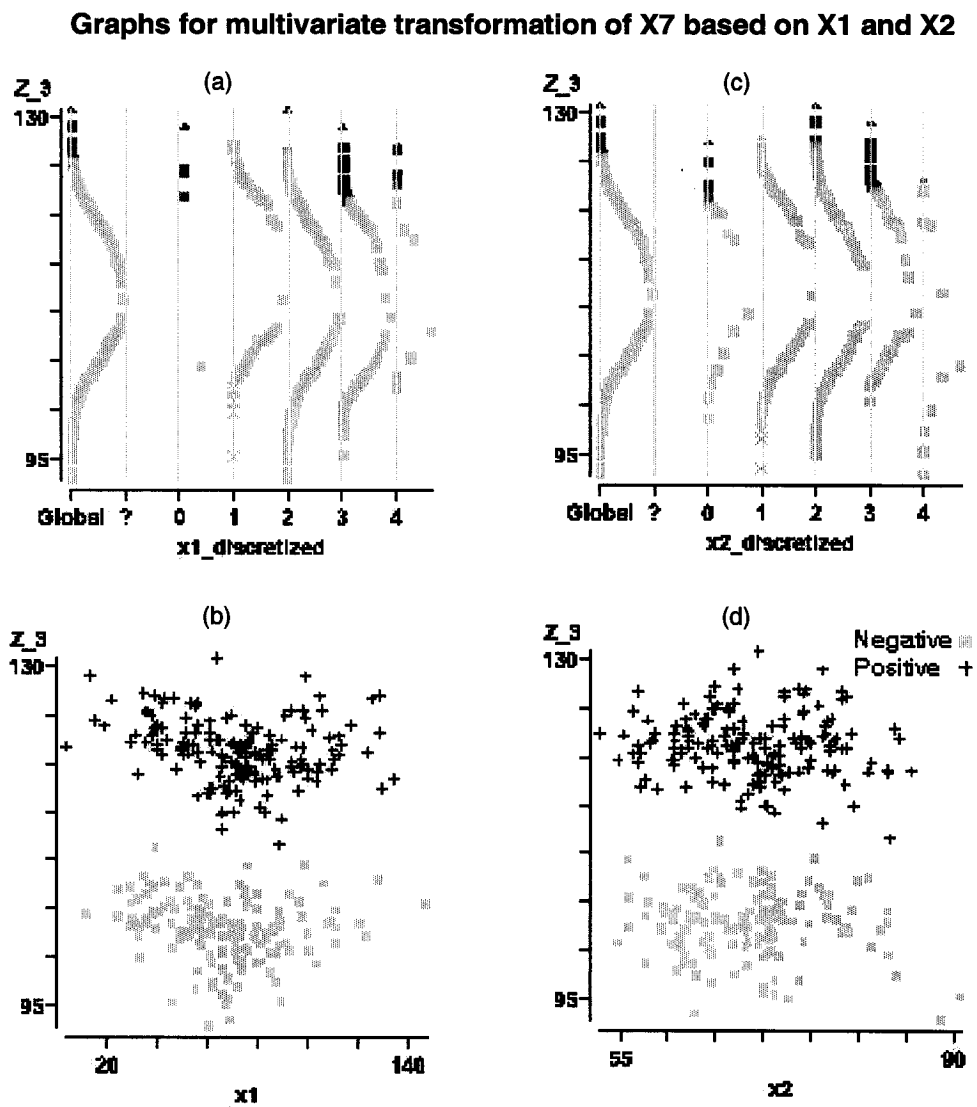


Figure 6.13: Relevance graphs and scatter plots showing the relations between the response attribute X_7 and the two explanatory attributes X_1 and X_2 in domain N5MN.

the dependency since the initial relationship between X_7 and X_2, shown in Figure 6.10, is no longer noticeable. However, the effects of the three methods on class separability are very different. The PCA technique had no positive repercussion, the ICA technique helped somewhat whereas the GLOREF approach greatly increased the separability by producing a near perfect solution.

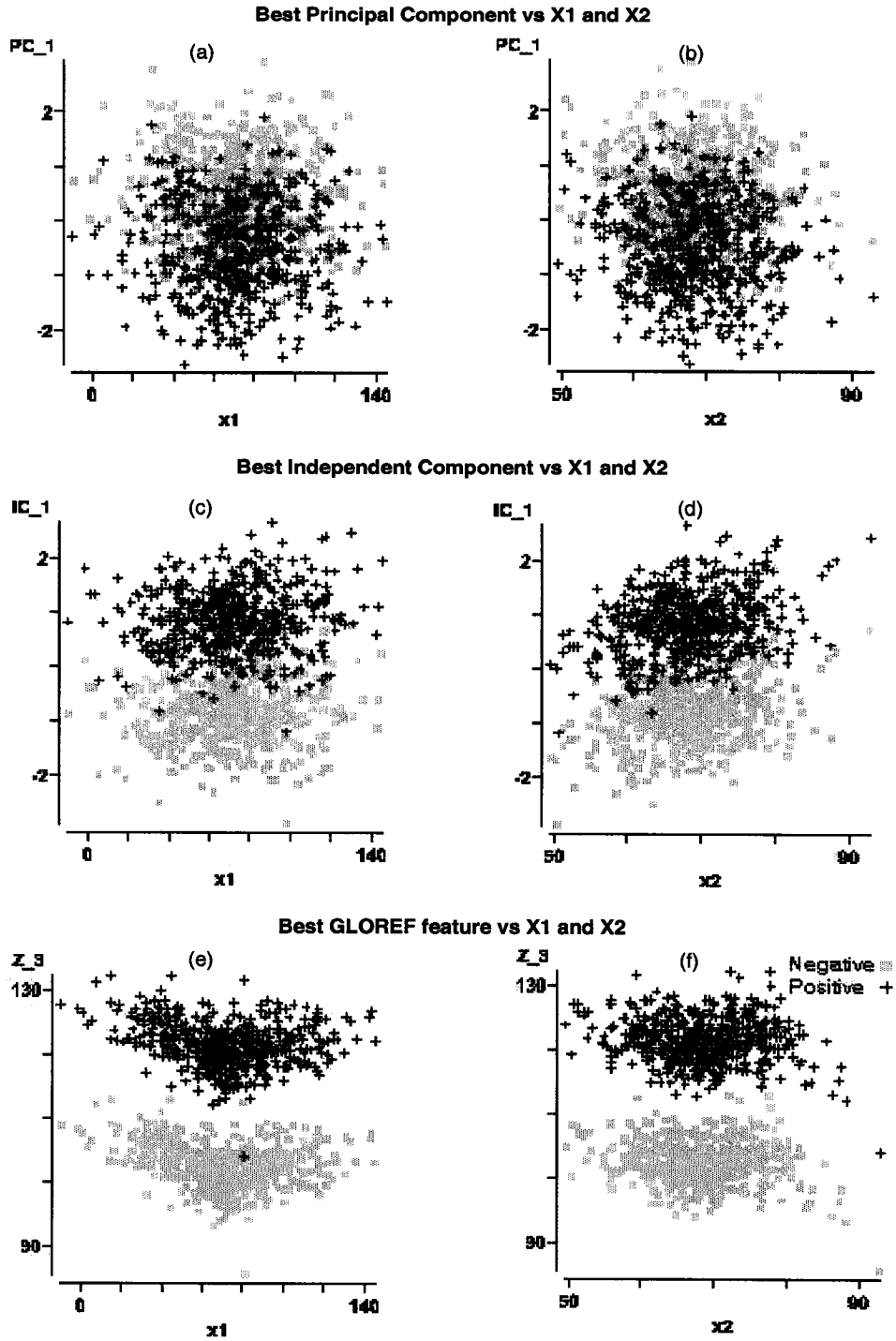


Figure 6.14: Scatter plots of the best PCA, ICA, and GLOREF features versus the explanatory attributes X_1 and X_2 from domain N5MN.

6.7 Limitations

This section discusses issues with the proposed experimental study and limitations of the GLOREF approach.

6.7.1 Human involvement and interaction with the relevance graphs

All experiments involved in this chapter have been automated. This automation made it possible to investigate the feasibility of the GLOREF approach in a variety of domains with a rigorous methodology. On the other hand, by automating the process we disregarded the potential benefits of an important contribution of this research: the relevance graphs, presented in the second half of Chapter 4.

One of the advantages of the relevance graphs is to allow the researcher to converge towards a solution by potentially avoiding many non-essential steps. For instance, in the example considered in the previous section, a quick look at the relevance graphs among the initial attributes would have indicated the need for a transformation between X_7 and X_2. Following this step, the need to jointly consider X_1 would have become clear. After considering the excellent performance obtained by transforming X_7 for the joint effects of X_1 and X_2, the researcher would have probably decided to interrupt the search for other transformations. This means that we could have limited the search for univariate transformations to two pairs of attributes (X_7 versus X_1 and X_7 versus X_2) as opposed to the 36^3 pairs for the automatic approach. Moreover, many computationally expensive multivariate optimizations would have been avoided.

Allowing the user to interact with the relevance graphs might also increase the likelihood of detecting anomalies during the analysis. For instance, the GLOREF method is expected to be sensitive to the discretization rules used to partition the data based on numerical explanatory attributes (Section 4.2.3). Moreover, the classical discretization algorithms used in machine learning research are not necessarily optimal for the GLOREF approach. In our experiments, we tried to minimize the risk of using a inappropriate discretization schema by systematically discretizing all numerical attributes with two methods (equal-width and equal-frequency (Section 4.2.3)). The consequence of this choice is an exponential increase in the number of pairwise interactions to consider during the search for univariate transformations (Section 5.2). Moreover, there is no guarantee that either of these two methods was appropriate for the domains considered, nor can we guarantee that the number of bins selected was optimal. Accordingly, although we obtained highly positive results with

[3] the number of pairs from the initial 9 attributes

a majority of the datasets, we are not sure that any of these results are optimal. In a more typical Data Mining application in which the user is interested to build models for a given dataset (and not for 24 datasets), it would be desirable to use the relevance graphs to confirm the appropriateness of the discretization used. This may help improve the results further.

Finally, the GLOREF approach can make use of incomplete domain knowledge about attribute interactions. For instance, instead of proceeding with a search for all pairs of response-explanatory attributes, the user could provide to the system an initial list of attributes that are likely to interact. This allows the user to efficiently test initial hypotheses about the relationships in the data. Furthermore, since the search size is exponential in the number of attributes, even a simple reduction in the number of attributes could greatly reduce the required time for the analysis. The current implementation allows the user to provide such a list of candidate attributes but the experimentation discussed in this chapter did not rely on this ability.

6.7.2 Time complexity

In Chapter 5, we discussed the complexity of the algorithms presented but the empirical evaluation presented in the current chapter is lacking information about running-time. The difficulty of extracting meaningful running-time information with the current implementation prevented us to perform an analysis of the running-time. The principal difficulty comes from the fact that we implemented the automatic approach as a wrapper around the system to visualize the relevance graphs. Consequently, during the searches for optimal sets of parameters, the system was continuously maintaining the complex data structures required to display the relevance graphs. Approximations obtained while profiling the system on small datasets suggested that around 60% of the time was spend in activities related to visualization. We are planning to decouple the search for transformations from the visualization functionality and then re-evaluate the running-time information.

The only time information measured during the experimentations is the total time required to process a given fold for all classifiers. This total time includes the following operations:

- Reading of the training and test data for the current fold
- Discretization
- Generation of the GLOREF features on the training data

- Apply feature selection to generate the various feature subsets
- Build the models for all classifiers and feature subsets
- Apply the GLOREF transformations to test data
- Evaluate the performance of all models built
- Save the results

Table 6.13 presents, for the artificial datasets, the total time information along with information on the size of the search performed by the GLOREF approach. Each line presents the results for one of the six dependency structures studied (Section 3.4). We have grouped the datasets by dependency structure to avoid duplication of the information[4]. The second column provides the number of initial attributes. The third column gives the average running-time per fold (covering all operations mentioned above). Finally, the fourth and fifth columns provide approximation for the numbers of univariate and multivariate transformations considered by the GLOREF approach. The number of univariate transformation is

$$num\_inital\_att * (2 * num\_inital\_att - 2) \tag{6.2}$$

This formula takes into account the fact that we generated two discretized attributes for each initial continuous attribute as well as the exclusion of pairs response-explanatory involving a single attribute (e.g., $x1$ vs $x1\_discretized1$ and $x1$ vs $x1\_discretized2$). To evaluate the number of multivariate transformations we considered the fact that all searches stopped after the second layer and we assumed that exactly 10 multivariate features were provided as input to the second layer. This last assumption complies with the average number of input features observed across all experiments with the artificial datasets. Therefore, the formula used to approximate the number of multivariate features is

$$C^2_{num\_univ\_features} + C^2_{10} \tag{6.3}$$

where C^r_n stands for the number of combinations of n features taken r at a time.

With the current implementation, we evaluate that the GLOREF related operations account for approximatively 50% of the total time spent in a given fold. The remaining 50% was required to perform the other operations mentioned above. Accordingly, to obtain the time for the GLOREF procedure alone, one would need to divide the running-times

[4]The difference in running times for the two datasets (functional and stochastic) for a given interaction structure happen to be very small.

Datasets	Num initial attributes	Avg time per fold (min)	Approx number of transformations	
			Univar	Multivar
n1f1 and n1mn	4	11.20	24	321
n2f1 and n2mn	4	8.89	24	321
n3f1 and n3mn	6	42.71	60	1815
n4f1 and n4mn	5	33.05	40	825
n5f1 and n5mn	9	61.38	72	2601
n6f1 and n6mn	9	50.45	72	2601

Table 6.13: Running-time information for the artificial datasets along with information about the complexity of the GLOREF search.

provided in Table 6.13 by two. This leads to running-times that are approximately 10 times slower than the C4.5 system for the same datasets. This is significantly faster than training a Neural-Network with the standard back-propagation algorithm. Nevertheless, as mentioned above, we plan to increase the performance by removing the dependencies in our implementation between the automatic search and the visualization module. Also, we plan to investigate a parallel version of the algorithm to exploit commodity computer clusters.

6.7.3 Adding versus replacing features

Throughout the dissertation, we focused on the *extension* of the initial representation by adding new globally relevant features to the initial attributes. An alternative approach is to replace the initial attributes by the new ones or replace some of them. Some recent constructive induction systems integrate the feature selection with the generation approach [Bloedorn, 1996; Hu, 1999]. The GLOREF approach also uses feature selection as an internal mechanism to control the size of the feature space when generating the multivariate features but never to the extent of removing any of the initial attributes.

The reason for not attempting to directly replace the initial attributes with the new features is simple. Our approach is meant to be an additive approach that adds power to the representation by addressing a very specific issue; the attribute interactions that reduce the relevance. As shown through various examples, we intentionally ignore certain types of interactions that do not deteriorate the relevance. Similarly, we do not attempt to exploit any information from the attributes that do not interfere with the global relevance of

the attributes. As a result, the new features produced by the GLOREF approach may not capture the totality of the information contained in the initial representation. Systematically removing all initial attributes would probably lead to important loss of useful information in many applications. The user might also leave some of the interactions and initial attributes intact, for their own reasons (e.g., because the attributes are important as they are in the business process). The proposed approach respects such choices by not forcing a replacement of the initial representation.

On the other hand, it is true that the GLOREF approach does not fully resolve the problem of building an appropriate representation for learning since some of the initial attributes might be irrelevant and the GLOREF approach do not remove them. This explains why all of the experiments performed included a feature selection step as part of the learning process. This additional step could have been avoided if the GLOREF solution would have completely handled the problem of data re-representation.

6.7.4 Types of interactions that cannot be handled by the proposed transformations

The results provided strong empirical evidences that the proposed transformations are useful in increasing the accuracy of many learning systems. On the other hand, one can find examples of interactions for which the proposed transformations are not optimal. We provided such an example in Section 4.3.3 (Figure 4.8). Figure 6.15 re-displays the example with additional details about compatibility information. We observe that the most relevant cut points from the two partitions are aligned on the same threshold (T=50), but they are incompatible. This causes a null global relevance. The best that can be done with the given set of transformations is to position the most relevant cut points in such a way that they do not interact anymore. Figure 6.16 illustrates such a transformation. Although we notice an improvement in global relevance, the transformation is not optimal.

A better alternative would be to resolve the incompatibility using a *mirror transformation* through a set of points. Formally, let us define a new feature Z produced by a mirror transformation as

$$Z = \Delta(X, Y; \Phi, \Psi) = \begin{cases} Y - \phi_1 \, \text{dist2}(\psi_1, Y) & \text{if } X = X(1) \\ Y - \phi_2 \, \text{dist2}(\psi_2, Y) & \text{if } X = X(2) \\ \vdots & \\ Y - \phi_3 \, \text{dist2}(\psi_m, Y) & \text{if } X = X(m) \end{cases} \tag{6.4}$$

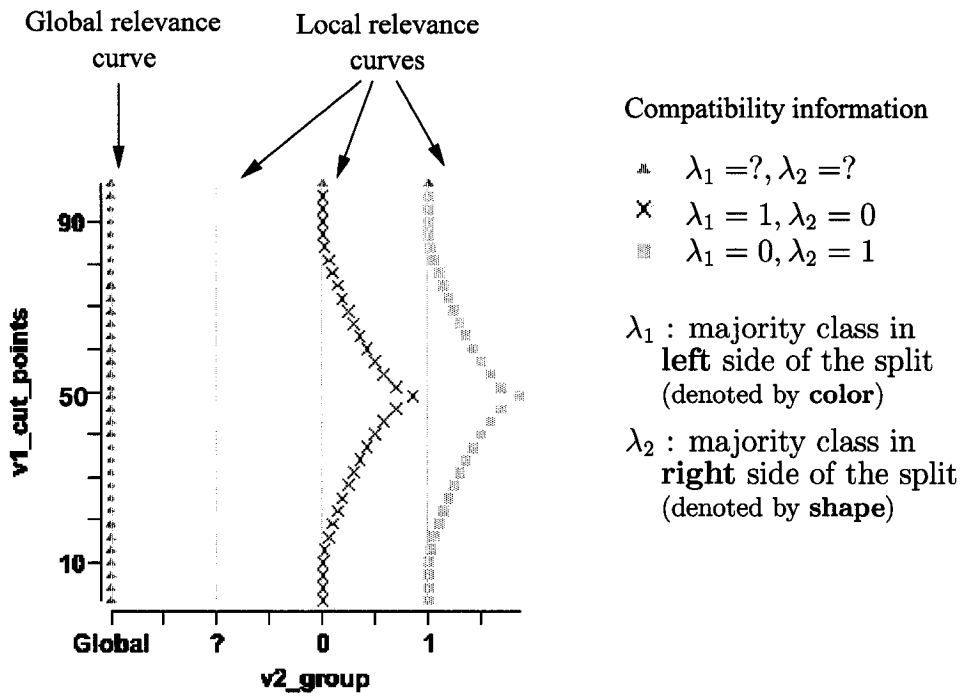


Figure 6.15: Example of an interaction where the most relevant cut points are incompatible.

where

$$\text{dist2}(\psi, Y) = 2(\psi - Y) \tag{6.5}$$

and the parameters $\Phi = \{\phi_1, \phi_2, \ldots, \phi_m\}$ and $\Psi = \{\psi_1, \psi_2, \ldots, \psi_m\}$ are real numbers.

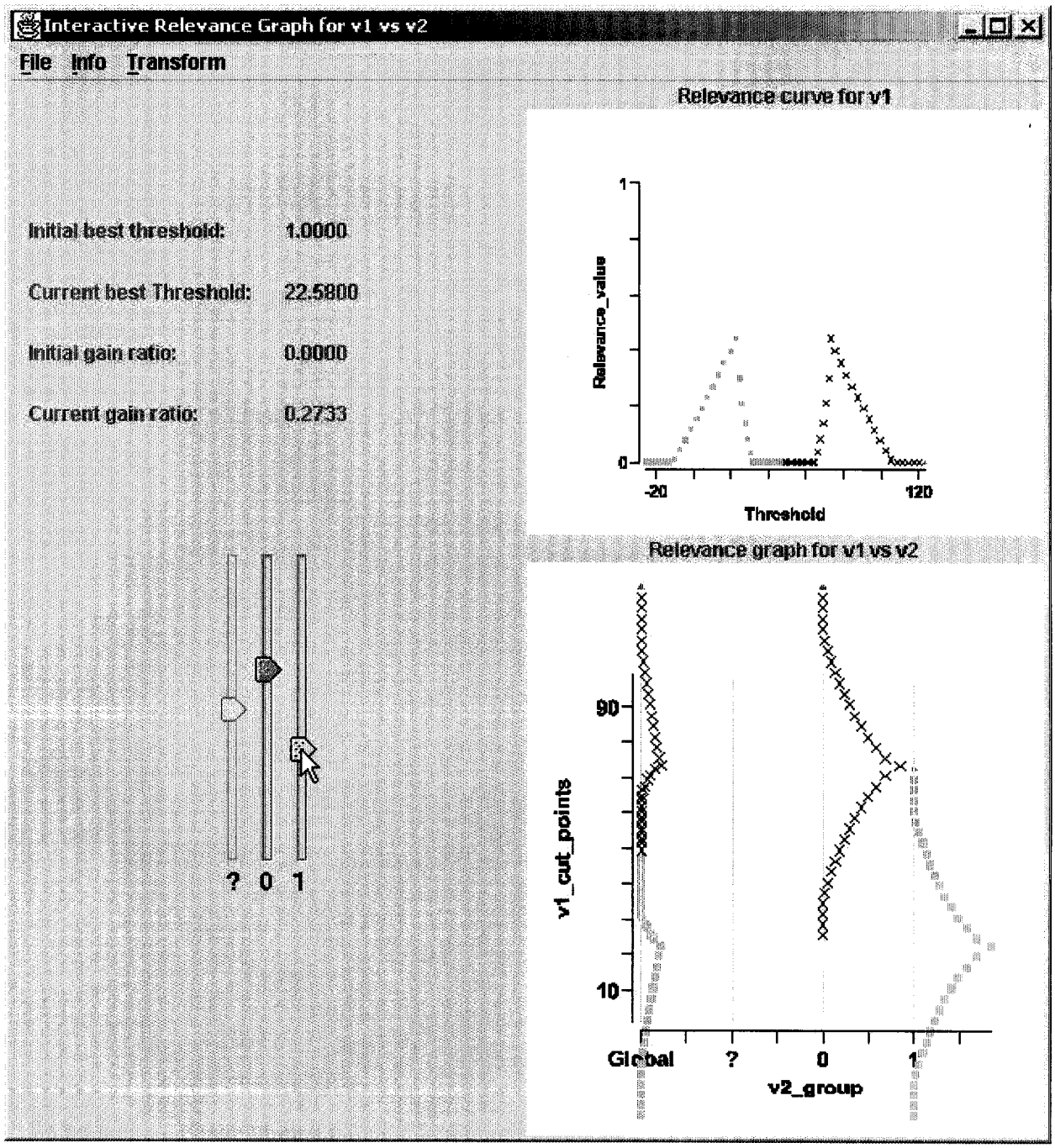


Figure 6.16: Best transformation that can be found by the GLOREF approach for the example presented in Figure 4.8.

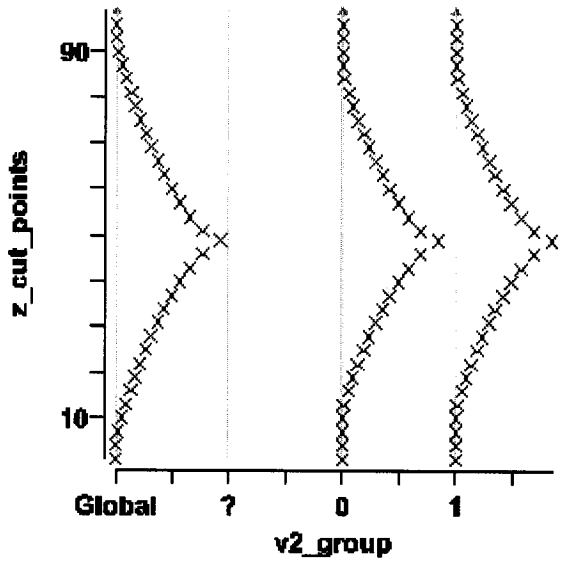


Figure 6.17: Illustration of the effects of the *mirror transformation* for the example presented in Figure 6.15.

To illustrate, let us define a new feature Z for the example above such that

$$Z = \Delta(v_2, v_1; 0.0, 1.0, 0.0, 50.0) \tag{6.6}$$

$$= \begin{cases} v_1 - 0.0\,\text{dist2}(0.0, Y) & \text{if } v_2 = \text{'0'} \\ v_1 - 1.0\,\text{dist2}(50.0, Y) & \text{if } v_2 = \text{'1'} \end{cases} \tag{6.7}$$

This mirror transformation does not affect the values in the first group (v_2 ='0') but causes a *flipping* of all values in the second group (v_2 ='1') around the threshold 50. This flipping inverts the distribution characteristics λ_1 and λ_2 for all points in the second group. As a results, all points are now compatible across the two groups (Figure 6.17). This leads to an optimal global relevance.

The example above illustrates that the GLOREF approach is not optimal with respect to all kinds of interactions. However, one needs to recognize that adding support for additional transformations, such as the mirror transformation defined above, requires a thorough analysis since any such additions may introduce other complications (e.g., How to adapt the search procedure to multiple forms of transformations? What would be the behavior in multi-class problems? How to handle missing values? How the new transformation would

interfere with the smoothing operation? What would be the impact on the performance?) These are important issues that could be addressed in future work to enhance the power of the GLOREF approach.

6.8 Summary

This chapter discussed a set of experiments to evaluate the feasibility of the GLOREF approach. We first described the methodology. The experiments covered the synthetic datasets introduced in Chapter 3 and 12 real-world datasets from the UCI repository. We analyzed the global relevance of the GLOREF features using the gain ratio and χ^2 measures. According to both measures, the GLOREF approach succeeded in generating globally relevant continuous features. Adding the new features to the initial data representation significantly increased the accuracy for a large number of domains and classifiers. We compared the results obtained with the GLOREF features and the results obtained using the theoretical features discussed in Chapter 3. Overall, the results obtained with the GLOREF features are comparable to the ones obtained with the theoretical features. This demonstrates that the GLOREF approach can compete with features derived from perfect knowledge of the attribute interactions. We also compared the results with two other general pre-processing techniques (PCA and ICA). The GLOREF features largely surpassed the PCA features in improving the accuracy. The GLOREF features also appeared superior to the ICA features. We presented a case study in which we illustrated, through a series of relevance graphs and scatter plots, the successive transformations that lead to one of the globally relevant features produced by the GLOREF approach. We completed the chapter by discussing the limitations of the GLOREF approach and the experiments performed.

Chapter 7

Conclusion

This final chapter has 3 sections. The first one offers a summary of the dissertation, the second one discusses the main contributions, and the third one suggests areas for future research.

7.1 Summary

This research is about improving the performance of learning systems on datasets characterized by attribute interactions. In Chapter 1, we used two examples to show how attribute interactions may reduce the global relevance of the initial attributes. For each example, we proposed a transformation that removes the negative effects of the initial interactions. These transformations illustrated the potential benefits of globally relevant features. We provided the context for this research by proposing a classification of possible solutions to address the problem of learning with attribute interactions. We listed the contributions of the dissertation and presented the thesis statement:

> *Augmenting the data representation with globally relevant continuous features, derived from the analysis of interactions between nominal and continuous attributes (or only between continuous attributes), improves the performance of several learning systems.*

Chapter 2 reviewed related work. Following the classification structure introduced in the first chapter, the review of literature covered two types of solutions: enhancements of existing learning approaches independent of the data representation, and constructive induction techniques. The discussions on enhancements proposed outside the area of constructive induction included advanced feature selection mechanisms in decision trees as well as extensions to the naive-Bayes classifier. Regarding the constructive induction approaches, we first reviewed the related work on contextual normalization. Then, we presented several constructive induction systems according to the role of attribute interactions during the search for new features. Our review of constructive induction also included techniques that

have been developed in the related fields of pattern recognition and statistics. The analysis of existing techniques showed a lack of pre-processing methods that can generate continuous attributes optimized for classification tasks. We also noticed that the current tools are very limited with respect to user interactions.

Chapter 3 commenced the investigation of the thesis statement by proposing an idealized experimental study to evaluate the potential benefits of globally relevant features. The study is based on 12 synthetic datasets with pre-determined types of attribute interactions. We explained two general techniques that we have developed to generate synthetic datasets with functional and stochastic attribute interactions given any dependency structures specified by a belief network. Assuming perfect knowledge of the interactions, we explained how to compute globally relevant features for such data. We named these new features *theoretical features* to emphasize the fact that they are derived from perfect knowledge of the initial attribute interactions. We evaluated the potential benefits of the theoretical features by comparing the accuracy of models learned from the initial representation and models learned using an extended representation that includes the new features. The accuracies were evaluated through a rigorous methodology. The results obtained demonstrated that globally relevant features can significantly improve the accuracy with all of the 11 learning systems considered. They also indicated that the use of globally relevant features is not likely to reduce the accuracy. Chapter 3 concluded that the main challenge of this research was to develop methods to derive globally relevant features for datasets for which the attribute interactions are unknown.

We developed the GLOREF approach as a general technique to create globally relevant features for datasets with unknown interactions. The GLOREF approach consists of two phases. The first phase concentrates on the analysis of interactions along with their effects on the relevance of the attributes. The second phase uses the information computed during the analysis of relevance to efficiently generate globally relevant features. These two phases were described in Chapters 4 and 5, respectively.

Chapter 4 introduced two examples to show that some attribute interactions may appear identical for classical statistical techniques and yet, affect the learning problem at hand quite differently. In particular, one of the examples was clearly increasing the complexity of the learning task, while the other was indeed simplifying the learning. Classical statistical techniques were unable to distinguish the two interactions because they do not take into account the class information. To overcome the weakness of current techniques, we proposed a method that analyses attribute interactions in terms of their effects on the relevance of the attributes involved in the interactions. We described the computations required and

the *relevance matrices* which are used to store the results of the analysis of relevance.

The second part of Chapter 4 presented the *Relevance graph*, the visualization technique we have invented to visualize the content of the relevance matrices. We have explained the components of a relevance graph which are the relevance curves and the annotations of compatibility. We have also described the theory developed to support the interpretation of the relevance graphs. Finally, we presented various scenarios to illustrate the use of relevance graphs in practical applications.

The last part of Chapter 4 described the *Interactive relevance graph* tool which allows the researcher to explore possible transformations to cancel the negative effects of the interactions observed. We discussed the functionalities of the tool and showed how it can be used to create a globally relevant feature for one of the examples presented in the introduction of this chapter.

Chapter 5 completed the presentation of the GLOREF approach by explaining the generation of globally relevant features. In introduction, we offered a schematic view of the GLOREF approach. This clarified the strong links between the relevance matrices computed in Chapter 4 and the generation of globally relevant features. The second section introduced the mathematical models used to generate univariate and multivariate globally relevant features and the corresponding optimization problems. The following two sections presented the solution proposed for the univariate and multivariate problems, respectively. The presentation of the solution for the univariate case included the analysis of the problem space, the exhaustive solution, and various algorithms developed to ensure adequate performance. We provided several examples to illustrate the usefulness of the proposed algorithms. For the multivariate case, we suggested two approaches. The first one extends the univariate solution by applying a multivariate partitioning procedure. After explaining the limitations of this solution, we described a second approach based on an inductive process. This approach considers multivariate transformations in progressive order of complexity. The approach exploits the univariate models and optimization algorithms already developed, which allows an efficient resolution of the complex optimization problem. We explained the risk of generating transformations that overfit the data and described the technique we used to control this risk. We also discussed the alternative stopping criteria and other parameters that can be introduced to create variations of the proposed approach. The last part of Chapter 5 discussed the application of the transformation models. We have proposed solutions for three important practical issues: missing values, unseen values, and smoothing of transformations involving continuous explanatory attributes.

To fulfill the objective of the thesis statement, we needed to show that:

1. The GLOREF approach is capable of generating globally relevant continuous features.
2. Augmenting the data representation with the new features produced by the GLOREF approach increases the accuracy of existing learning systems.

We verified the above requirements in Chapter 6 through an extensive empirical evaluation of the GLOREF approach. After explaining the methodology, we presented the results on the 12 synthetic datasets and 12 real-world datasets, using 13 classifiers.

For the synthetic datasets, the analysis of the relevance of the new features showed that the GLOREF system generated highly globally relevant features for all datasets. In some domains, we observed increases in gain ratio that were close to 500%. As expected, the increases in relevance were less important in the datasets with weaker interactions. Adding the GLOREF features to the initial representation significantly improved the accuracy in more than 70% of the experiments. Moreover, the extent of improvement in accuracy was directly related to the gains observed in global relevance. All of the 13 learning systems significantly improved their performance when allowed to use the GLOREF features. As anticipated, the most important improvements were noticed with the simplest learning systems such as OneR, HyperPipes, and NaiveBayes which are all well-known for their strong reliance on the assumption of attribute independence. Also remarkable is the fact that among the 132 cross-validation experiments with the synthetic data, we observed only 2 cases for which the addition of the GLOREF features significantly reduced the accuracy.

The results with the real world datasets were also strongly positive. First, both the gain ratio and χ^2 measures confirmed that the GLOREF approach generated features that were significantly more relevant than the best initial attribute in 11 datasets out of 12. The difference in global relevance for the remaining dataset was not significant. Among the 114 experiments performed, the GLOREF features statistically significantly improved the accuracy in 27 experiments ($\approx$24%) while reducing it in only 2 experiments ($\approx$1.7%). In summary, the results with both the synthetic and real world datasets forcefully show that the GLOREF approach complies with the two requirements mentioned above. Consequently, we have empirically demonstrated the validity of the thesis statement.

In the second part of Chapter 6, we compared GLOREF with alternative approaches and presented a case study. First, we contrasted the GLOREF features with the theoretical features discussed in Chapter 3. Overall, the results obtained with the GLOREF features are comparable to the ones obtained with the theoretical features. This demonstrates that the GLOREF approach can compete with features derived from perfect knowledge of

the attribute interactions. Second, we compared the results with two other general preprocessing techniques (PCA and ICA). The GLOREF features largely surpassed the PCA features in improving the accuracy. The GLOREF features were also superior to the ICA features. The case study presented a series of relevance graphs and scatter plots that showed the successive transformations that lead to one of the globally multivariate relevant features produced by the GLOREF approach. The exact mathematical transformations found by the GLOREF system were also presented.

Finally, the last part of Chapter 6 discussed the issues with the experimental study and limitations of the GLOREF approach.

7.2 Contributions

This dissertation succeeded in demonstrating that globally relevant continuous features can enhance the performance of learning systems. The research accompanying this achievement offers the following contributions:

A method and a large-scale experiment to study the effects of attribute interactions in learning This contribution comes from Chapter 3. As we explained in the related work section of this chapter, previous studies provided only an indirect evaluation of the effects of attribute interactions by comparing classifiers on real-world datasets. Since the characteristics of the interactions in these datasets used were unknown, the conclusions from these studies were limited to statements like: *Learning system S1 appears to be less affected than learning system S2.* They were also limited in scope since they considered a relatively small number of classification systems. The method we introduced provides a direct evaluation leading to statements like: *Attribute interactions I appear to decrease/increase the accuracy of learning system S by an amount A*, where the characteristics of the interactions I are formally defined through a belief network and an interaction type (functional or stochastic, in the current implementation), and where any standard learning systems S can be used. The approach proposed is also generic and can be easily extended to experiments with other types of attribute interactions (e.g., other belief networks, other functional or stochastic relationships). Moreover, the results of the experiment performed increase our knowledge about the effects of attribute interactions and help us better understand experimental results provided by other researchers. In particular, we now have empirical evidences that systems such as decision trees and support vector machines, often considered reliable against attribute interactions, are also negatively affected by various types of interactions.

This explains why previous comparative studies only observed minor differences between the performance of the naive-Bayes and decision tree approaches.

A novel method to assess attribute interactions that takes into account the effects on learning Previous techniques to assess attribute interactions involving numerical attributes are not taking the class information into account and/or are limited to continuous attributes only. As shown in Section 4.2.2, ignoring the class information may lead to misleading information by suggesting transformations that would increase the complexity of the learning task. The limitation to only continuous attributes is also problematic since most real world datasets contain a mix of continuous and nominal attributes (e.g., see Table 6.5 which details the UCI datasets used). The method for the analysis of relevance, described in Section 4.2, resolves both problems since it handles mixed attributes and links the analysis of interactions to their effects on the learning problem at hand. Moreover, the proposed technique is well complemented by the relevance graphs and the automatic feature generation approach, which are two other important contributions.

An interactive tool to visualize the effects of interactions and to explore possible transformations The literature review has shown that existing constructive induction methods do not usually involve the user in the analysis of interactions. This prevents the analysis method to exploit the potentially important information that the user could provide, and also limits the quality of the experience for the user. These are important constraints in complex applications in which the user typically has some knowledge of the data and the desire to further extend his/her own understanding of the effects of the interactions among the attributes. The *Relevance Graphs*, introduced in Section 4.3, address this weakness by allowing the user to participate in the analysis of interactions. With the relevance graphs, we have introduced a theory that allows a precise interpretation of the graphs. As we illustrated in Section 4.3.3, the system we have developed to display relevance graphs allows efficient identification of several forms of interactions along with their impact on the learning problem at hand. The complementary system named *Interactive Relevance Graph* fills another important need by providing the user with the opportunity to participate in the construction of a new representation that cancels the effects of the negative attribute interactions. The software we have developed to implement these two novel interactive techniques have been used extensively throughout this research. We believe that they represent noticeable additions to the set of tools available to researchers in Machine Learning and Knowledge Discovery in Databases.

A new data transformation technique that generates globally relevant continuous features The GLOREF approach which integrates the analysis of relevance and a new automatic procedure to generate globally relevant continuous features is probably the most important contribution of this research. As described in Chapter 5, the GLOREF approach integrates many new algorithms and optimization procedures. The proposed approach is complex but very powerful since it can handle univariate and multivariate interactions in an efficient manner. As explained in Section 5.4, the GLOREF approach also includes a number of additional mechanisms to deal with important issues in real world applications such as missing or unknown values. By implementing all the mechanisms described, we obtained a powerful system which we used for the large-scale experiments presented in the Sections 6.2 to 6.6. The results from these experiments, the comparisons with other techniques, and the case study presented unambiguously demonstrate the power and the usefulness of the GLOREF approach.

Early work on this research appeared in Létourneau et al. [1998]. This work has been largely motivated by real-world applications we described in Létourneau et al. [1997]; Delisle et al. [1998]; Létourneau [Létourneau]; Peña, Létourneau & Famili [Peña et al.]; Peña, Famili & Létourneau [Peña et al.], and Létourneau et al. [1999].

7.3 Future research

This section presents future work to address some of the limitations discussed in Section 6.7 and foster the cohesion of research in constructive induction.

7.3.1 Using general optimization algorithms

The approaches introduced in this dissertation to resolve the univariate and multivariate optimization problems (Section 5.1.1) rely on task specific heuristics. These have been shown efficient and robust over the various datasets we have experimented with. However, there would be advantages in using a general optimization approach to resolve these two problems. For instance, there are good generic libraries available for optimization problems and using them may simplify the implementation. Using such library may also facilitate the extension of the GLOREF framework to work with other types of transformations. The fields of mathematical programming and constraint satisfaction problems offer general techniques that could be used to solve the optimization problems defined in this dissertation. It is not clear that these techniques would lead to improved performance but their integration

is worth considering.

7.3.2 Parallelizing the GLOREF approach

Powerful clusters of computers, which are increasingly available, could be used to speed up the GLOREF approach. This is particularly promising since GLOREF's most expensive operations in terms of computations would be relatively easy to parallize. Specifically, the three most important operations to consider are:

Computation of the relevance matrices As we have seen in Chapter 4, a group of relevance matrices is computed for each pair of explanatory and response attributes. The computations between pairs of attributes are independent and thus, they could be performed on different computing nodes (or CPUs). The procedure would be as follows. During initialization, the master process would transmit to each computing node the values for the explanatory, response, and class attributes. After computations of the relevance matrices, the computing nodes would return the relevance matrices to the master nodes. Ideally, the computing nodes would also calculate the additional information required to generate the relevance graphs (i.e., the relevance information and the cut points distribution characteristics, Section 4.3). This would allow the master node to generate the relevance graphs without further computation. The benefits of this parallel procedure over the sequential approach would depend on the difference between the time required to compute the relevance matrices and the communication time required to transmit the initial values. Given the speed of most networks available today, it may happen that the communications take longer than the computation of the values in the relevance matrices. If that is the case, it would be best not to parallize this step. The next two operations are more promising.

Computation of the univariate transformations The GLOREF approach computes a univariate transformation for each pair of explanatory and response attributes by resolving the optimization problem defined by Equation (5.3). The various univariate transformation problems can be processed on different computing nodes since they are independent of each other. As we explained in Section 5.2, the only required input is the set of relevance matrices corresponding to the given pair of attributes. The output is a relatively small set of real values (the values for $\{\omega_1, \omega_2, \ldots, \omega_m\}$). Accordingly, the communication overhead caused by the parallization would be relatively low. On the other hand, the optimization process is complex and requires the evaluation of

many candidate solutions. Although each potential solution can be evaluated efficiently using the algorithms proposed, the overall search may still take a considerable amount of time. In this case, we anticipate that the communication overhead would represent a small fraction of the time required by the optimization process. Therefore, we believe that the parallization of the computations of the univariate transformations would be near optimal (i.e., parallel time $\approx$ sequential time divided by number of CPUs).

Computation of the multivariate transformations The most expensive operations of the inductive approach introduced in Section 5.3.2 happen in the feature generation step (Figure 5.8). This step solves the multivariate optimization problem, defined by Equation (5.5), for all combinations of features from the previous phase. The approach can be parallized by distributing the computations for these combinations over the various computing nodes of a cluster of computers. As with the univariate transformations, the communication overhead would be limited to transmissions of a few relevance matrices and the set of real values describing the obtained solution. On the other hand, the search process is even more complex than in the univariate case. Accordingly, we expect that the parallization of the computations of the multivariate transformations would be near optimal.

7.3.3 A larger framework that integrates with other data transformation techniques

Finally, techniques to improve data representation tend to be developed as stand-alone tools. It would be useful to have a general framework that could be used to completely specify the existing techniques. The recent framework proposed by Markovitch & Rosenstein [2002] is definitely a step in this direction. However, it is not clear how the GLOREF approach and other statistical techniques would fit in the proposed framework. We believe that this kind of work should be encouraged to tighten the links between research in constructive induction and to facilitate the implementation of novel and existing techniques. Ultimately, this should increase the role of constructive induction techniques in machine learning and should help tackle successfully complex applications.

Bibliography

Aha, D. W. & Kibler, D. F. (1991). Instance-based learning algorithms. *Machine Learning, 6*, 37–66.

Bloedorn, E. & Michalski, R. S. (1996). The AQ17-DCI system for data-driven constructive induction and its application to the analysis of world economics. In *Proceedings of the 9th International Symposium, ISMIS '96*, (pp. 108–117).

Bloedorn, E. & Michalski, R. S. (1998). Data-driven constructive induction. *IEEE Intelligent Systems and their Applications, 13*(2), 30–37.

Bloedorn, E., Michalski, R. S., & Wnek, J. (1993). Multistrategy constructive induction: AQ17-MCI. In *Proceedings of the 2nd International Workshop on Multi-Strategy Learning.*

Bloedorn, E. E. (1996). *Multistrategy Constructive Induction.* PhD thesis, Departement of Computer Science and System Engineering, George Masson University, Fairfax, Virginia.

Breiman, L. Bagging predictors", journal = Machine Learning, volume = 24, number = 2, pages = 123–140, year = 1996.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and Regression Trees.* Belmont, CA: Wadsworth International Group.

C. Blake, E. K. & Merz, C. (1998). UCI repository of machine learning databases.

Cheng, J. & Greiner, R. (1999). Comparing Bayesian network classifiers. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence (UAI-99).* Morgan Kaufmann.

Clark, P. & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In Y. Kodratoff (Ed.), *Machine Learning - EWSL-91* (pp. 151–163). Springer-Verlag.

Clark, P. & Niblett, T. (1988). The CN2 induction algorithm. *Machine Learning, 3*, 261.

Codrington, C. W. & Brodley, C. E. (1999). On the qualitative behavior of impurity-based splitting rules: The minima-free property. Technical report, School of Electrical and Computer Engineering, Purdue University.

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

Cristianini, N. & Shawe-Taylor, J. (2000). *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.* Cambridge University Press.

Delbeke, K., Teklemariam, T., de la Cruz, E., & Sorgeloos, P. (1995). Reducing variability in pollution data: The use of lipid classes for normalization of pollution data in marine biota. *International Journal of Environmental Analytical Chemistry, 58*(1), 147–162.

Delisle, S., Létourneau, S., & Matwin, S. (1998). Experiments with learning parsing heuristics. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, (pp. 307–314).

Devijver, P. & Kittler, J. (1982). *Pattern recognition: a statistical approach.* London: Prentice Hall.

Dietterich, T. G. (1998). Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation, 10*(7), 1895–1923.

Dietterich, T. G., London, B., Clarkson, K., & Dromey, G. (1982). Learning and inductive inference. In Cohen, P. R. & Feigenbaum, E. A. (Eds.), *The Handbook of Artificial Intelligence: Volume III*, (pp. 323–512). William Kaufman, Inc.

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

Donoho, S. & Rendell, L. A. (1996). Constructive induction using fragmentary knowledge. In *Proceedings of the 13th International Conference on Machine Learning*, (pp. 113–121). Morgan Kaufmann.

Donoho, S. K. (1996). *Knowledge-Guided Constructive Induction.* PhD thesis, Computer Science, University of Illinois at Urbana-Champaign. Report no. UIUCDCS-R-96-1970.

Donoho, S. K. & Rendell, L. A. (1995). Lessons from theory revision applied to constructive induction. In *Proceedings of the 12th International Conference on Machine Learning*, (pp. 185–193). Morgan Kaufmann.

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. In *International Conference on Machine Learning*, (pp. 194–202).

Duda, R. O. & Hart, P. E. (1973). *Pattern Classification and Scene Analysis.* New York: John Wiley & Sons.

Elio, R. & Watanabe, L. (1991). An incremental deductive strategy for controlling constructive induction in learning from examples. *Machine Learning, 7*, 7–44.

Elomaa, T. & Rousu, J. (1999). General and efficient multisplitting of numerical attributes. *Machine Learning, 36*(3), 201–244.

Famili, A. F., Shen, W.-M., Weber, R., & Simoudis, E. (1997). Data preprocessing for intelligent data anaysis. *International Journal of Intelligent Data Analysis.*

Fawcett, T. (1993). *Feature Discovery for Problem Solving Systems.* PhD thesis, Computer Science, University of Massachusetts.

Fayyad, U. & Irani, K. (1992). On the handling of continuous-valued attributes in decision tree generation. *Machine Learning, 8*, 87–102.

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1995). *Advances in Knowledge Discovery and Data Mining.* The MIT Press.

Fayyad, U. M., Weir, N., & Djorgovski, S. (1993). Automated cataloging and analysis of sky survey image databases: the SKICAT system. In *Proceedings of the 2nd International Conference on Information and Knowledge Management*, (pp. 527–536).

Fraser, D. (1978). *Probability and Statistics: Theory and Applications.* North Scituate, MA: Duxbury Press.

Freitas, A. A. (2001). Understanding the crucial role of attribute interaction in data mining. *Artificial Intelligence Review, 16*, 177–199.

Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *International Conference on Machine Learning*, (pp. 148–156).

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning, 29*, 131–161.

Fukunaga, K. (1990). *Introduction to Statistical Pattern Recognition, 2nd edition.* NY: Academic Press.

Harries, M., Sammut, C., & Horn, K. (1998). Extracting hidden context. *Machine Learning, 32*, 101–126.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. *Machine Learning, 11*, 63–91.

Hong, S. J. (1997). Use of contextual information for feature ranking and discretization. *IEEE Transactions on Knowledge and Data Engineering, 9*(5), 718–730.

Horrigan, J. (1978). *Financial Ratio Analysis: An Historical Perspective.* New York: Arno Press.

Hu, Y.-J. (1999). *Representational Transformation Through Constructive Induction.* PhD thesis, University of California, Irvine.

Hyvärinen, A., Karhunenand, J., & Oja, E. (2001). *Independent Component Analysis: Algorithms and Applications.* New York: John Wiley & Sons, Inc.

Iba, W. & Langley, P. (1992). Induction of one-level decision trees. In Kaufmann, M. (Ed.), *Proceedings of the Ninth International Conference on Machine Learning*, (pp. 233–240).

Isaaks, E. H. & Srivastava, R. M. (1989). *An Introduction to Applied Geostatistics.* New York, Oxford: Oxford Univ. Press.

Jakulin, A. (2002). Attribute interactions in machine learning. Master's thesis, University of Ljubljana.

John, G. H., Kohavi, R., & Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *Proc. 11th International Conference on Machine Learning*, (pp. 121–129). Morgan Kaufmann.

Johnson, M. E. (1987). *Multivariate Statistical Simulation.* New York: Wiley.

Johnson, R. & Wichern, D. (1992). *Applied Multivariate Statistical Analysis.* Prentice Hall.

Kass, G. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, *29*, 119–127.

Katz, A. J., Gatley, M. T., & Collins, D. R. (1990). Robust classifiers without robust features. *Neural Computation*, *2*, 472–479.

Kerber, M. (1992). ChiMerge: Discretization of numeric attributes. In *Proceedings of the 10th National Conference on Artificial Intelligence*, (pp. 123–128). AAAI Press.

Kira, K. & Rendell, L. A. (1992). A practical approach to feature selection. In Sleeman, D. & Edwards, P. (Eds.), *Proceedings of the 9th International Conference on Machine Learning*, (pp. 249–256). Morgan Kaufmann Publishers.

Knutson, L., Soderberg, G., Ballantyne, B., & Clarke, W. (1994). A study of various normalization procedures for within day electromyographic data. *Journal of Electromyography and Kinesiology*, *4*(1), 47–59.

Kohavi, R. (1995). *Wrappers for Performance Enhancement and Oblivious Decision Graphs.* PhD thesis, Stanford University.

Kohavi, R. (1996). Scaling up the accuracy of Naive-Bayes classifiers: a decision-tree hybrid. In Simoudis, E., Han, J. W., & Fayyad, U. (Eds.), *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, (pp. 202–207). AAAI Press.

Kokar, M. M. (1986). Determining arguments of invariant functional descriptors. *Machine Learning*, *1*(4), 403–422.

Kononenko, I. (1991). Semi-naive Bayesian classifier. In Kodratoff, Y. (Ed.), *Proceedings of the European Working Session on Learning : Machine Learning (EWSL-91)*, volume 482 of *LNAI*, (pp. 206–219). Springer Verlag.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of RELIEF. In Bergadano, F. & de Raedt, L. (Eds.), *Proceedings of the European Conference on Machine Learning*, volume 784 of *LNAI*, (pp. 171–182). Springer-Verlag.

Kononenko, I. & Hong, S. J. (1997). Attribute selection for modelling. *Future Generation Computer Systems*, *13*, 181–195.

Kononenko, I. & Šimec, E. (1995). Induction of decision trees using ReliefF. In G. Della Riccia, R. Kruse, & R. Viertl (Eds.), *Mathematical and Statistical Methods in Artificial Intelligence, CISM courses and lectures No. 363.* Udine, Italy: Springer Verlag.

Kramer, S. (1994). CN2-MCI: A two-step method for constructive induction. In *Proceedings of the 11th International Conference on Machine Learning (ICML-94)*, (pp. 33–39).

Kubat, M. (1996). Second tier for decision trees. In *Proceedings of the 13th International Conference on Machine Learning*, (pp. 293–301).

Kvålseth, T. (1987). Entropy and correlation: Some comments. *IEEE Trans. on Systems, Man and Cybernetics, SMC-17(3)*, 517–519.

Langley, P. (1993). Induction of recursive Bayesian classifier. In *Proceedings of the European Conference on Machine Learning*, (pp. 152–164). Springer-Verlag.

Langley, P., Bradshaw, G. L., & Simon, H. A. (1986). Rediscovering chemistry with the Bacon system. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning, an Artificial Intelligence approach*, volume 2 (pp. 307–329). San Mateo, California: Morgan Kaufmann.

Langley, P. & Sage, S. (1994). Induction of selective Bayesian classifiers. In *Proceedings of the tenth Conference on Uncertainty in Artificial Intelligence*, (pp. 399–406). Morgan Kaufmann.

Langley, P. & Simon, H. (1995). Applications of machine learning and rule induction. *Communication of the ACM*, *38*(11), 54–64.

Lee, S.-W. & Park, J.-S. (1994). Nonlinear shape normalization methods for the recognition of large-set handwritten characters. *Pattern Recognition*, *27*(7), 895–902.

Lenat, D. B. & Brown, J. S. (1984). Why AM and EURISKO appear to work. *Artificial Intelligence*, *23*(3), 269–294.

Létourneau, S. Data mining for maintenance of complex systems. In *AAAI/IAAI 1998*, (pp. 1178).

Létourneau, S., Famili, A. F., & Matwin, S. (1997). Discovering useful knowledge from aircraft operation/maintenance data. In *Proceedings of the workshop on Machine Learning Applications in the Real World, 14th International Conference on Machine Learning*, (pp. 34–41).

Létourneau, S., Famili, A. F., & Matwin, S. (1998). A normalization method for contextual data: Experience from a large-scale application. In *Proceedings of the 10th European Conference on Machine Learning (ECML-98)*, (pp. 49–54).

Létourneau, S., Famili, A. F., & Matwin, S. (1999). Data mining for prediction of aircraft component replacement. *IEEE Intelligent Systems and their Applications*, *14*(6), 59–66.

Liu, H., Hussain, F., Tan, C. L., & Dash, M. (2002). Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, *6*(4), 393–423.

Liu, H. & Motoda, H. (1998). *Feature Selection for Knowledge Discovery and Data Mining.* Kluwer Academic Publishers.

Markovitch, S. & Rosenstein, D. (2002). Feature generation using general constructor functions. *Machine Learning*, *49*(1), 59–98.

Matheus, C. (1989). *Feature Construction: An Analytical Framework and an Application to Decision Trees.* PhD thesis, Computer Science, University of Illinois at Urbana-Champaign.

Matheus, C. & Rendell, L. A. (1989). Constructive induction on decision trees. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI-89)*, (pp. 645–650).

Matwin, S. & Kubat, M. (1996). The role of context in concept learning. In *Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains*, (pp. 1–5).

Michalski, R. S. (1983). A theory and methodology of inductive learning.

Michalski, R. S., Mozetic, I., Hong, J., & Lavrac, N. (1986). The multi-purpose incremental learning system AQ15 and its testing application to three medical domains. In *Proceedings of the 5th National Conference on Artificial Intelligence*, (pp. 1041–1045).

Mingers, J. (1989). An empirical comparison of selection measures for decision tree induction. *Machine Learning*, *3*, 319–342.

Mizzaro, S. (1997). Relevance: The whole history. *Journal of the American Society of Information Science*, *48*(9), 810–832.

Murthy, S. K., Kasif, S., & Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, *2*, 1–33.

Murthy, S. K. & Salzberg, S. (1995). Lookahead and pathology in decision tree induction. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI-95).*

Pagallo, G. (1990). *Adaptive Decision Tree Algorithms for Learning from Examples.* PhD thesis, University of California at Santa Cruz.

Pagallo, G. & Haussler, D. (1990). Boolean feature discovery in empirical learning. *Machine Learning, 5*(1), 71–99.

Pazzani, M. J. (1995). Searching for dependencies in Bayesian classifiers. *Proceedings of the 5th International Workshop on Artificial Intelligence and Statistics*, 239–248.

Pazzani, M. J. (1996). Constructive induction of Cartesian product attributes. *ISIS: Information, Statistics and Induction in Science*, 66–77.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* San Mateo, California: Morgan Kaufmann.

Peña, J. M., Famili, F., & Létourneau, S. Data mining to detect abnormal behavior in aerospace data. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, (pp. 390–397).

Peña, J. M., Létourneau, S., & Famili, F. Application of rough sets algorithms to prediction of aircraft component failure. In *Advances in Intelligent Data Analysis: Third International Symposium, IDA-99*, (pp. 473–486).

Pérez, E. & Rendell, L. A. (1996a). Learning despite concept variation by finding structure in attribute-based data. In *Proceedings of the 13th International Conference on Machine Learning*, (pp. 391–199). Morgan Kaufmann.

Pérez, E. & Rendell, L. A. (1996b). Statistical variable interaction: Focusing multiobjective optimization in machine learning. In *First International Workshop on Machine Learning, Forecasting and Optimization*, Madrid, Spain.

Pfahringer, B. (1994a). Controlling constructive induction in CIPF: An MDL approach. In Bergadano, F. & Raedt, L. D. (Eds.), *Proceedings of the 7th European Conference on Machine Learning (ECML-94)*, (pp. 242–256). Springer-Verlag.

Pfahringer, B. (1994b). Robust constructive induction. In *Lecture notes in computer science*, volume 861, (pp. 118–129).

Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. In R. S. Michalski, J. G. Carbonell, & T. M. Mitchell (Eds.), *Machine Learning, an Artificial Intelligence approach*, volume 1 (pp. 463–482). San Mateo, California: Morgan Kaufmann.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*, 81–106.

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

Quinlan, J. R. & Rivest, R. L. (1989). Inferring decision trees using the Minimum Description Length Principle. *Information and Computation, 80*(3), 227–248.

Rakotomalala, R. (1997). *Graphes d'induction.* PhD thesis, Université Claude Bernard - LYON I.

Rendell, L. A. & Ragavan, H. (1993). Improving the design of induction methods by analyzing algorithm functionality and data-based concept complexity. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence IJCAI-93*, (pp. 952–958).

Rendell, L. A. & Seshu, R. (1990). Learning hard concepts through constructive induction: Framework and rationale. *Computational Intelligence, 6*(4), 247–270.

Richeldi, M. & Lanzi, P. L. (1996). Performing effective feature selection by investigating the deep structure of the data. In Simoudis, E., Han, J. W., & Fayyad, U. (Eds.), *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, (pp. 379–382). AAAI Press.

Robnik-Šikonja, M. & Kononenko, I. (1997). An adaptation of Relief for attribute estimation in regression. In *Proceedings of the 14th International Conference on Machine Learning*, (pp. 296–304). Morgan Kaufmann.

Sahami, M. (1996). Learning limited dependence Bayesian classifiers. In Simoudis, E., Han, J. W., & Fayyad, U. (Eds.), *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, (pp. 335). AAAI Press.

Schlimmer, J. (1987). Learning and representation change. In *Proceedings of the National Conference on Artificial Intelligence*, (pp. 511–515). AAAI Press.

Sled, J., Zijdenbos, A., & Evans, A. (1998). A nonparametric method for automatic correction of intensity nonuniformity in MRI data. *IEEE Transactions on Medical Imaging, 17*(1), 87–97.

Sola, J. & Sevilla, J. (1997). Importance of input data normalization for the application of neural networks to complex industrial problems. *IEEE Transactions on Nuclear Science, 44*(3), 1464–1468.

Toth, M., Goran, M., Ades, P., Howard, D., & Poehlman, E. (1993). Examination of data normalization procedures for expressing peak VO2 data. *Journal of Applied Physiology, 75*(5), 2288–2292.

Turney, P. (1993). Exploiting context when learning to classify. In *Proceedings of the European Conference on Machine Learning (ECML-93)*, (pp. 402–407).

Turney, P. (1996a). The identification of context-sensitive features: A formal definition of context for concept learning. In *Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains*, (pp. 53–59).

Turney, P. (1996b). The management of context-sensitive features: A review of strategies. In *Proceedings of the ICML-96 Workshop on Learning in Context-Sensitive Domains*, (pp. 53–59).

Turney, P. & Halasz, M. (1993). Contextual normalization applied to aircraft gas turbine engine diagnosis. *Applied Intelligence, 3*, 109–129.

Utgoff, P. & Brodley, C. (1990). An incremental method for finding multivariate splits for decision trees. In *Proceedings of the 7th International Conference on Machine Learning*, (pp. 58–65). Morgan Kaufmann.

Utgoff, P. E. (1986). Shift of bias for inductive concept learning. In J. G. Carbonell, R. S. Michalski, & T. M. Mitchell (Eds.), *Machine Learning: An Artificial Intelligence Approach, Volume II* (pp. 107–148). Los Altos, CA: Morgan Kaufmann.

Vilata, R., Blix, G., & Rendell, L. A. (1997). Global data analysis and the fragmentation problem in decision tree induction. (pp. 312–326).

Watanabe, S. (1969). *Knowing and guessing - A formal and quantitative study.* New York: John Wiley & Sons, Inc.

Watrous, R. (1991). Context-modulated vowel discrimination using connectionist networks. *Computer Speech and Language, 5*, 341–362.

Watrous, R. (1993). Speaker normalization and adaptation using second-order connectionist networks. *IEEE Transactions on Neural Networks, 4*, 21–30.

Widmer, G. (1996). Recognition and exploitation of contextual clues via incremental meta-learning. In *Proceedings of the 13th International Conference on Machine Learning*, (pp. 525–533).

Widmer, G. & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning, 23*(1), 69–101.

Widmer, G. & Kubat, M. (1998). Guest editors' introduction. *Machine Learning, 32*, 83–84.

Witten, I. H. & Frank, E. (1999). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations.* Morgan Kaufmann.

Wnek, J. & Michalski, R. S. (1994). Hypothesis-driven constructive induction in AQ17-HCI: A method and experiments. *Machine Learning, 14*(2), 139–168.

Yang, D., Rendell, L. A., & Blix, G. (1991). A scheme for feature construction and comparison of empirical methods. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, (pp. 699–704).

Yip, S. P. & Webb, G. I. (1994). Incorporating canonical discriminant attributes in classification learning. In *Proceedings of the 10th Biennial Conference of the Canadian Society for Computational Studies of Intelligence*, (pp. 63–70).

Zembowicz, R. & Zytkow, J. (1991). Automated discovery of empirical equations from data. In Zemankova, R. (Ed.), *Proceedings of 6th International Symposium on Methodologies for Intelligent Systems (ISMIS-91)*, (pp. 429–440). Springer Verlag.

Zheng, Z. (1996). *Constructing New Attributes for Decision Tree Learning.* PhD thesis, Basser Department of Computer Science, The University of Sydney.

Appendix A

Description of learning systems used

All the classification systems used to evaluate the GLOREF approach are part of the WEKA software. The Java source code is available from `http://www.cs.waikato.ac.nz/ml/weka/`. We used the version 3.2 of WEKA with default settings for all classification systems. This appendix provides a short description of the classifiers used with references to relevant papers. An introduction to most of these classifiers can be found in the book complementing the WEKA software [Witten & Frank, 1999].

BaggingDT A bagging classifier using J48 as weak classifier [Breiman, Breiman].

BoostingDT A boosting classifier based on Freund and Schapire's Adaboost M1 method [Freund & Schapire, 1996]. The weak classifier is J48.

DecisionStump A one-level decision tree [Iba & Langley, 1992].

DecisionTable A simple classifier based on a partition defined over a selected subset of attributes and the majority rule [Kohavi, 1995].

HyperPipes A simple classifier for which the learning phase is reduced to recording, for each class, the domains of all the attributes. The HyperPipe is the structure that stores all the domains for a given class. For a continuous attribute, the domain is defined by the minimal and maximal values observed for this attribute over all instances from the given class. For a nominal attribute, the domain is defined by the list of values observed for this attribute over all instances from the given class. To classify a new instance, the system evaluates the *coverage* of each HyperPipe by counting the number of domains that contain the instance's attribute values. The new instance is assigned to the class associated with the HyperPipe having the largest coverage value.

IB1 An instance-based learning system that predicts the class value of a new instance based on *the* closest instance [Aha & Kibler, 1991]. This is equivalent to a *k-nearest-neighbors* classifier with $k = 1$.

IB5 An instance-based learning system that predicts the class value of a new instance based on the closest *5* instances [Aha & Kibler, 1991]. This is equivalent to a *k-nearest-neighbors* classifier with $k = 5$.

J48 A Java implementation of the C4.5 system [Quinlan, 1993].

KernelDensity A simple kernel density classifier based on a Gaussian kernel [Duda & Hart, 1973].

NaiveBayes A classifier based on Bayes's rules with the assumption of attribute independence [Duda & Hart, 1973]. This implementation evaluates the probabilities for continuous attributes by assuming that they are normally distributed.

OneR A simple classifier that uses a single attribute to predict the class [Holte, 1993].

PART A rule-based classifier generated from a pruned-decision tree (from J48) [Witten & Frank, 1999].

SMO A support vector machines classifier based on the *sequential minimal optimization* algorithm and a *polynomial kernel* of degree n (n=1, by default) [Cristianini & Shawe-Taylor, 2000].

Appendix B

Tabulated results for Chapter 3

Table B.1: Results for dataset **N1F1** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	95.2 ±3.39	95.3 ±3.47	95.8 ±1.69	**96.8 ±1.14**	1.50
BoostingDT	95.8 ±1.55	95.2 ±1.69	96.1 ±1.79	**96.2 ±1.62**	0.40
DecisionStump	83.0 ±5.56	83.0 ±5.56	**88.4 ±3.20**	**88.4 ±3.20**	5.40*
DecisionTable	86.2 ±4.61	86.2 ±4.61	94.9 ±2.60	**95.2 ±2.39**	9.00*
HyperPipes	56.4 ±3.06	56.1 ±3.14	**70.3 ±4.24**	68.8 ±4.44	13.9*
IB1	95.7 ±2.45	95.4 ±2.59	96.0 ±1.83	**96.1 ±1.66**	0.40
IB5	96.2 ±2.10	95.8 ±2.66	96.3 ±1.64	**96.8 ±1.03**	0.60
J48	92.3 ±4.47	92.0 ±4.81	95.2 ±2.39	**95.9 ±1.91**	3.60*
KernelDensity	96.0 ±2.40	95.5 ±2.80	96.2 ±1.93	**96.7 ±1.49**	0.70
NaiveBayes	87.6 ±3.50	88.0 ±3.59	95.5 ±2.80	**97.4 ±1.17**	9.40*
OneR	81.1 ±5.51	81.1 ±5.51	**86.1 ±2.64**	**86.1 ±2.64**	5.00*
PART	93.5 ±3.03	92.9 ±3.48	**95.4 ±2.27**	95.2 ±1.75	1.90
SMO	96.9 ±1.97	96.7 ±2.16	**97.8 ±1.23**	97.7 ±1.06	0.90

Table B.2: Results for dataset **N1MN** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	78.0 ±4.35	79.3 ±4.57	**79.5 ±5.04**	79.0 ±6.06	0.20
BoostingDT	77.0 ±6.67	77.5 ±4.86	76.1 ±5.04	**78.1 ±5.67**	0.60
DecisionStump	65.5 ±3.81	65.3 ±3.65	**76.4 ±3.37**	**76.4 ±3.37**	10.9*
DecisionTable	78.0 ±5.21	74.4 ±5.76	**81.4 ±4.40**	81.3 ±3.53	3.40
HyperPipes	54.2 ±1.75	53.5 ±1.65	**57.7 ±2.58**	55.5 ±2.32	3.50*
IB1	74.2 ±3.36	70.7 ±3.56	75.9 ±3.45	**76.3 ±3.06**	2.10
IB5	77.9 ±4.68	75.7 ±3.50	80.5 ±3.34	**81.2 ±2.39**	3.30
J48	75.8 ±3.22	75.7 ±2.58	80.3 ±4.83	**81.2 ±3.39**	5.40*
KernelDensity	76.8 ±4.49	75.7 ±3.95	77.0 ±3.30	**79.0 ±2.62**	2.20
NaiveBayes	79.9 ±4.56	76.2 ±5.73	82.2 ±5.96	**83.8 ±4.80**	3.90
OneR	65.0 ±4.94	65.3 ±4.85	**70.6 ±3.81**	**70.6 ±3.81**	5.30*
PART	76.2 ±3.88	75.7 ±3.59	80.6 ±2.80	**81.0 ±3.37**	4.80*
SMO	80.0 ±4.47	78.2 ±3.39	**83.2 ±4.26**	**83.2 ±4.24**	3.20

Table B.3: Results for dataset **N2F1** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	96.6 ±1.58	96.4 ±2.32	**97.2 ±1.32**	**97.2 ±1.62**	0.60
BoostingDT	96.5 ±1.78	96.3 ±2.06	**97.9 ±1.37**	96.6 ±1.43	1.40
DecisionStump	**93.8 ±2.57**	**93.8 ±2.57**	**93.8 ±2.57**	**93.8 ±2.57**	0.00
DecisionTable	**96.4 ±1.96**	**96.4 ±1.96**	95.9 ±1.66	95.4 ±1.58	−.50
HyperPipes	90.3 ±2.91	89.9 ±3.38	**90.8 ±2.66**	89.4 ±3.75	0.50
IB1	**97.1 ±1.37**	96.4 ±1.65	96.9 ±1.45	96.7 ±1.57	−.20
IB5	**97.7 ±1.49**	97.3 ±1.70	**97.7 ±1.25**	96.6 ±2.17	0.00
J48	**96.8 ±1.62**	96.2 ±1.55	96.6 ±1.43	95.3 ±2.26	−.20
KernelDensity	**97.2 ±1.55**	96.9 ±1.91	97.0 ±1.63	96.8 ±1.93	−.20
NaiveBayes	97.7 ±0.95	97.6 ±1.43	**97.9 ±1.29**	96.4 ±2.41	0.20
OneR	**93.3 ±2.75**	**93.3 ±2.75**	**93.3 ±2.75**	**93.3 ±2.75**	0.00
PART	**96.3 ±1.83**	95.9 ±2.18	95.2 ±1.81	96.0 ±2.05	−.30
SMO	96.8 ±1.03	96.9 ±1.20	**97.3 ±1.25**	96.1 ±1.97	0.40

Table B.4: Results for dataset **N2MN** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	88.1 ±3.11	**89.3 ±3.80**	88.2 ±3.16	87.7 ±4.32	−1.1
BoostingDT	**90.0 ±2.58**	89.0 ±5.73	87.7 ±4.42	87.4 ±4.06	−2.3
DecisionStump	67.4 ±2.80	67.4 ±2.80	**91.9 ±1.85**	**91.9 ±1.85**	24.5*
DecisionTable	79.9 ±3.67	80.0 ±3.80	**94.5 ±2.27**	94.1 ±2.33	14.5*
HyperPipes	52.9 ±1.66	52.9 ±1.66	**76.7 ±3.80**	75.8 ±3.82	23.8*
IB1	82.7 ±3.37	80.7 ±4.72	**94.0 ±2.16**	93.8 ±3.65	11.3*
IB5	84.0 ±2.21	83.5 ±2.37	**96.3 ±1.77**	95.9 ±1.85	12.3*
J48	82.6 ±2.22	83.2 ±2.86	**94.3 ±2.36**	93.8 ±2.78	11.1*
KernelDensity	84.5 ±3.17	84.2 ±2.70	94.2 ±2.35	**94.8 ±3.08**	10.3*
NaiveBayes	80.6 ±2.95	80.0 ±3.30	**95.1 ±0.99**	95.0 ±1.70	14.5*
OneR	66.1 ±2.02	66.1 ±2.02	**91.8 ±2.35**	**91.8 ±2.35**	25.7*
PART	81.4 ±2.12	81.8 ±1.93	**94.1 ±2.02**	93.7 ±2.63	12.3*
SMO	84.7 ±3.86	84.7 ±3.86	**95.9 ±2.02**	95.7 ±1.77	11.2*

Table B.5: Results for dataset **N3F1** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	98.5 ±1.08	98.2 ±1.75	**98.7 ±0.67**	98.1 ±1.10	0.20
BoostingDT	**99.0 ±1.05**	98.5 ±2.07	98.7 ±0.82	98.2 ±1.14	−.30
DecisionStump	91.5 ±2.32	91.5 ±2.32	**93.8 ±2.44**	**93.8 ±2.44**	2.30*
DecisionTable	97.0 ±1.56	97.1 ±1.60	**97.8 ±1.55**	97.0 ±2.26	0.70
HyperPipes	87.8 ±4.08	87.9 ±4.12	**94.8 ±2.25**	93.7 ±3.06	6.90*
IB1	98.3 ±1.64	98.2 ±1.14	**98.6 ±1.35**	**98.6 ±2.07**	0.30
IB5	98.6 ±1.07	98.3 ±1.06	**99.2 ±0.63**	99.1 ±1.52	0.60
J48	96.5 ±1.65	97.0 ±1.15	**98.0 ±1.05**	97.7 ±1.64	1.00
KernelDensity	98.3 ±1.64	98.4 ±1.35	98.7 ±1.34	**98.8 ±1.81**	0.40
NaiveBayes	98.0 ±1.05	98.1 ±1.10	99.1 ±0.74	**99.2 ±1.32**	1.10
OneR	90.8 ±3.36	90.8 ±3.36	92.7 ±2.58	**92.9 ±2.51**	2.10
PART	97.4 ±1.58	96.5 ±1.18	**98.3 ±1.16**	98.0 ±1.83	0.90
SMO	99.2 ±0.63	98.4 ±0.84	**99.5 ±0.71**	99.1 ±1.29	0.30

Table B.6: Results for dataset **N3MN** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	**84.8 ±3.58**	83.7 ±3.27	84.3 ±4.00	84.5 ±2.80	−.30
BoostingDT	82.3 ±2.75	82.6 ±2.76	**83.0 ±3.16**	82.7 ±2.67	0.40
DecisionStump	72.1 ±4.53	72.1 ±4.53	**76.2 ±3.39**	**76.2 ±3.39**	4.10*
DecisionTable	77.6 ±4.06	77.6 ±4.06	82.2 ±3.19	**82.7 ±2.87**	5.10*
HyperPipes	50.9 ±0.99	50.9 ±0.99	**54.2 ±2.35**	52.5 ±1.78	3.30*
IB1	76.8 ±3.39	76.9 ±3.51	**79.8 ±3.55**	79.1 ±3.41	2.90
IB5	81.4 ±3.41	81.2 ±3.43	**84.0 ±3.09**	83.2 ±2.97	2.60
J48	77.9 ±3.90	78.0 ±3.83	81.7 ±3.65	**82.2 ±3.65**	4.20*
KernelDensity	77.1 ±3.75	77.0 ±3.62	79.9 ±3.45	**80.8 ±5.05**	3.70
NaiveBayes	80.9 ±4.75	80.9 ±4.75	85.4 ±3.66	**85.8 ±3.19**	4.90*
OneR	62.6 ±6.64	62.6 ±6.64	**75.5 ±6.29**	**75.5 ±6.29**	12.9*
PART	79.4 ±2.91	79.4 ±2.91	**83.7 ±2.11**	83.3 ±3.43	4.30*
SMO	**87.1 ±2.77**	86.7 ±3.47	86.8 ±3.26	86.4 ±2.50	−.30

Table B.7: Results for dataset **N4F1** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	92.9 ±2.64	92.9 ±2.18	**99.5 ±0.53**	99.4 ±0.70	6.60*
BoostingDT	95.8 ±2.66	95.5 ±2.17	**99.6 ±0.52**	99.2 ±0.79	3.80*
DecisionStump	79.6 ±3.47	79.6 ±3.47	**99.2 ±0.63**	**99.2 ±0.63**	19.6*
DecisionTable	83.8 ±3.05	83.8 ±3.05	**99.2 ±0.63**	**99.2 ±0.63**	15.4*
HyperPipes	59.7 ±3.27	59.7 ±3.27	99.2 ±1.48	**99.3 ±0.95**	39.6*
IB1	95.7 ±3.20	95.6 ±2.72	98.5 ±0.97	**99.5 ±0.71**	3.80*
IB5	95.2 ±2.49	95.1 ±2.42	99.0 ±1.15	**99.4 ±0.70**	4.20*
J48	92.3 ±2.11	92.2 ±2.10	**99.6 ±0.52**	99.3 ±0.67	7.30*
KernelDensity	96.6 ±2.12	96.3 ±1.95	98.5 ±0.97	**99.2 ±1.03**	2.60*
NaiveBayes	84.8 ±3.29	85.0 ±3.23	**99.8 ±0.42**	**99.8 ±0.42**	14.8*
OneR	79.4 ±3.78	79.4 ±3.78	**99.0 ±0.82**	**99.0 ±0.82**	19.6*
PART	91.9 ±2.02	91.3 ±2.91	**99.6 ±0.52**	99.3 ±0.67	7.70*
SMO	97.4 ±1.90	97.1 ±1.97	**99.2 ±0.63**	98.5 ±1.08	1.80*

Table B.8: Results for dataset **N4MN** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	94.5 ±2.46	94.7 ±2.50	**99.8 ±0.42**	**99.8 ±0.42**	5.10*
BoostingDT	95.8 ±1.55	96.3 ±2.00	**99.6 ±0.52**	**99.6 ±0.52**	3.30*
DecisionStump	66.0 ±3.23	66.0 ±3.23	**98.9 ±1.20**	**98.9 ±1.20**	32.9*
DecisionTable	78.5 ±1.78	78.5 ±1.78	**99.5 ±0.97**	**99.5 ±0.97**	21.0*
HyperPipes	52.0 ±1.41	52.2 ±1.40	98.4 ±1.26	**98.6 ±1.35**	46.4*
IB1	97.6 ±1.65	97.8 ±1.48	99.8 ±0.42	**100 ±0.00**	2.20*
IB5	99.0 ±1.33	99.2 ±1.14	99.9 ±0.32	**100 ±0.00**	0.80*
J48	87.0 ±2.79	87.0 ±2.79	99.6 ±0.52	**99.7 ±0.48**	12.7*
KernelDensity	97.7 ±1.57	98.3 ±1.34	99.8 ±0.42	**100 ±0.00**	1.70*
NaiveBayes	81.7 ±3.92	81.8 ±3.94	99.8 ±0.42	**99.9 ±0.32**	18.1*
OneR	66.7 ±4.62	66.7 ±4.62	**98.8 ±0.92**	**98.8 ±0.92**	32.1*
PART	88.5 ±3.63	89.1 ±3.03	99.7 ±0.48	**99.8 ±0.42**	10.7*
SMO	99.9 ±0.32	99.9 ±0.32	99.9 ±0.32	**100 ±0.00**	0.10

Table B.9: Results for dataset **N5F1** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	90.1 ±4.07	89.7 ±4.00	**96.3 ±2.31**	96.0 ±2.31	6.20*
BoostingDT	91.0 ±3.86	88.0 ±4.27	95.9 ±2.13	**96.2 ±2.15**	5.20*
DecisionStump	88.9 ±2.77	88.9 ±2.77	**92.1 ±3.07**	**92.1 ±3.07**	3.20*
DecisionTable	88.1 ±2.42	88.5 ±2.37	**96.3 ±1.89**	96.2 ±1.69	7.80*
HyperPipes	60.8 ±3.05	61.1 ±2.47	85.8 ±2.97	**86.2 ±2.57**	25.1*
IB1	83.6 ±2.22	84.3 ±3.92	94.0 ±1.83	**95.1 ±1.52**	10.8*
IB5	89.2 ±2.53	89.4 ±3.92	95.2 ±2.15	**96.6 ±1.90**	7.20*
J48	89.2 ±3.19	89.2 ±3.52	94.6 ±2.59	**95.8 ±2.10**	6.60*
KernelDensity	83.7 ±2.26	87.2 ±3.19	94.1 ±1.79	**95.4 ±1.35**	8.20*
NaiveBayes	89.6 ±2.80	89.5 ±3.92	96.5 ±2.07	**96.9 ±2.08**	7.30*
OneR	88.3 ±3.02	88.3 ±3.02	**91.9 ±2.81**	**91.9 ±2.81**	3.60*
PART	88.1 ±5.09	88.3 ±3.16	95.5 ±1.43	**95.8 ±1.93**	7.50*
SMO	94.5 ±3.06	90.6 ±4.84	**96.6 ±1.43**	96.4 ±1.58	2.10

Table B.10: Results for dataset **N5MN** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	90.2 ±4.52	91.7 ±3.59	**99.9 ±0.32**	**99.9 ±0.32**	8.20*
BoostingDT	94.2 ±2.49	95.0 ±3.43	**99.9 ±0.32**	**99.9 ±0.32**	4.90*
DecisionStump	70.8 ±3.68	68.5 ±3.75	**99.8 ±0.42**	**99.8 ±0.42**	29.0*
DecisionTable	77.9 ±6.35	73.5 ±8.53	**99.9 ±0.32**	**99.9 ±0.32**	22.0*
HyperPipes	51.1 ±2.13	50.9 ±1.91	**99.9 ±0.32**	84.6 ±20.0	48.8*
IB1	72.2 ±2.97	90.8 ±9.33	**99.9 ±0.32**	99.8 ±0.42	9.10*
IB5	90.7 ±3.06	93.9 ±5.59	**100 ±0.00**	99.9 ±0.32	6.10*
J48	83.6 ±4.50	88.6 ±3.60	**99.9 ±0.32**	**99.9 ±0.32**	11.3*
KernelDensity	72.6 ±2.84	91.8 ±8.32	**99.9 ±0.32**	**99.9 ±0.32**	8.10*
NaiveBayes	80.9 ±4.70	74.1 ±8.05	**100 ±0.00**	99.9 ±0.32	19.1*
OneR	64.5 ±3.27	63.0 ±3.89	**99.8 ±0.42**	**99.8 ±0.42**	35.3*
PART	85.0 ±4.64	90.5 ±2.76	**99.9 ±0.32**	**99.9 ±0.32**	9.40*
SMO	**100 ±0.00**	98.4 ±2.27	**100 ±0.00**	99.9 ±0.32	0.00*

Table B.11: Results for dataset **N6F1** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	88.8 ±4.76	89.0 ±4.55	**96.5 ±2.27**	96.0 ±2.62	7.50*
BoostingDT	89.2 ±4.61	88.1 ±4.68	**96.3 ±2.50**	96.2 ±2.15	7.10*
DecisionStump	81.0 ±5.14	81.0 ±5.14	**92.1 ±3.07**	**92.1 ±3.07**	11.1*
DecisionTable	81.0 ±5.03	80.8 ±5.22	96.3 ±1.89	**96.6 ±1.71**	15.6*
HyperPipes	52.3 ±1.25	52.0 ±1.25	85.2 ±3.26	**86.1 ±2.42**	33.8*
IB1	85.1 ±2.47	83.5 ±4.77	92.9 ±2.28	**94.6 ±2.63**	9.50*
IB5	87.2 ±3.43	86.6 ±4.84	95.5 ±2.32	**96.4 ±2.01**	9.20*
J48	87.5 ±4.33	86.3 ±4.14	95.2 ±2.53	**95.8 ±2.30**	8.30*
KernelDensity	85.4 ±2.46	84.9 ±3.60	92.9 ±2.28	**95.0 ±2.00**	9.60*
NaiveBayes	83.9 ±5.07	83.4 ±5.21	**97.2 ±1.48**	97.0 ±1.56	13.3*
OneR	79.3 ±4.42	79.3 ±4.42	**91.9 ±2.81**	**91.9 ±2.81**	12.6*
PART	85.4 ±3.92	85.7 ±4.22	95.5 ±1.90	**96.1 ±1.73**	10.4*
SMO	93.7 ±2.75	90.7 ±4.22	**96.7 ±1.57**	96.1 ±1.37	3.00*

Table B.12: Results for dataset **N6MN** with initial attributes and theoretical features.

Classifier	Initial attributes		Theoretical features		Improvement
	All	FS	All	FS	
BaggingDT	87.5 ±1.72	87.9 ±1.79	**91.6 ±2.95**	89.3 ±3.47	3.70*
BoostingDT	88.1 ±3.35	88.0 ±2.98	**91.0 ±3.86**	87.6 ±4.01	2.90
DecisionStump	67.3 ±4.69	67.3 ±4.69	**79.3 ±3.65**	**79.3 ±3.65**	12.0*
DecisionTable	78.7 ±2.91	77.5 ±2.80	89.0 ±3.68	**89.3 ±4.06**	10.6*
HyperPipes	53.1 ±2.08	53.3 ±1.89	**67.7 ±4.85**	67.3 ±4.57	14.4*
IB1	70.8 ±3.49	76.1 ±6.05	85.2 ±2.44	**86.5 ±2.80**	10.4*
IB5	82.0 ±2.83	83.0 ±4.16	**89.8 ±4.16**	89.2 ±3.65	6.80*
J48	79.6 ±2.76	79.6 ±3.44	88.3 ±3.37	**89.2 ±3.85**	9.60*
KernelDensity	71.0 ±3.40	76.6 ±5.46	85.2 ±2.49	**86.8 ±2.86**	10.2*
NaiveBayes	78.2 ±3.88	77.4 ±4.01	90.5 ±3.21	**91.8 ±3.77**	13.6*
OneR	63.9 ±5.45	63.9 ±5.45	**79.4 ±5.38**	**79.4 ±5.38**	15.5*
PART	82.1 ±4.86	80.0 ±3.71	**89.6 ±2.46**	88.9 ±3.35	7.50*
SMO	91.9 ±3.07	89.9 ±3.96	**92.4 ±3.10**	92.3 ±3.50	0.50

Appendix C

Tabulated results for Chapter 6

C.1 Results with GLOREF on Artificial datasets

Table C.1: Results for dataset **N1F1** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	95.2 ±3.39	95.3 ±3.47	95.4 ±3.37	**95.7 ±3.47**	95.8	96.8	0.40	−1.1
BoostingDT	95.8 ±1.55	95.2 ±1.69	**95.9 ±1.52**	95.7 ±2.00	96.1	96.2	0.10	−.30
DecisionStump	83.0 ±5.56	83.0 ±5.56	**94.2 ±2.90**	94.0 ±2.87	88.4	88.4	11.2*	5.80*
DecisionTable	86.2 ±4.61	86.2 ±4.61	**95.9 ±2.02**	95.5 ±1.84	94.9	95.2	9.70*	0.70
HyperPipes	56.4 ±3.06	56.1 ±3.14	**91.6 ±4.65**	79.0 ±7.83	70.3	68.8	35.2*	21.3*
IB1	95.7 ±2.45	95.4 ±2.59	95.5 ±2.59	**95.8 ±2.04**	96.0	96.1	0.10	−.30
IB5	**96.2 ±2.10**	95.8 ±2.66	95.5 ±2.64	96.1 ±1.73	96.3	96.8	−.10	−.70
J48	92.3 ±4.47	92.0 ±4.81	95.8 ±2.30	**95.9 ±2.13**	95.2	95.9	3.60*	0.00
KernelDensity	**96.0 ±2.40**	95.5 ±2.80	95.6 ±2.41	95.9 ±1.79	96.2	96.7	−.10	−.80
NaiveBayes	87.6 ±3.50	88.0 ±3.59	93.0 ±3.37	**95.7 ±3.65**	95.5	97.4	7.70*	−1.7
OneR	81.1 ±5.51	81.1 ±5.51	94.0 ±2.94	**94.4 ±3.17**	86.1	86.1	13.3*	8.30*
PART	93.5 ±3.03	92.9 ±3.48	94.9 ±3.28	**95.4 ±2.12**	95.4	95.2	1.90	0.00
SMO	96.9 ±1.97	96.7 ±2.16	**97.5 ±1.08**	96.5 ±2.72	97.8	97.7	0.60	−.30

Table C.2: Results for dataset **N1MN** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	78.0 ±4.35	79.3 ±4.57	**79.5 ±4.06**	78.1 ±5.97	79.5	79.0	0.20	0.00
BoostingDT	77.0 ±6.67	77.5 ±4.86	78.1 ±5.49	**78.7 ±5.54**	76.1	78.1	1.20	0.60
DecisionStump	65.5 ±3.81	65.3 ±3.65	**79.1 ±5.57**	79.0 ±5.16	76.4	76.4	13.6*	2.70
DecisionTable	78.0 ±5.21	74.4 ±5.76	78.4 ±5.34	**79.3 ±5.44**	81.4	81.3	1.30	−2.1
HyperPipes	54.2 ±1.75	53.5 ±1.65	**63.2 ±4.64**	55.4 ±2.77	57.7	55.5	9.00*	5.50*
IB1	**74.2 ±3.36**	70.7 ±3.56	73.4 ±4.93	73.3 ±3.77	75.9	76.3	−.80	−2.9
IB5	77.9 ±4.68	75.7 ±3.50	**78.6 ±3.66**	76.7 ±5.62	80.5	81.2	0.70	−2.6
J48	75.8 ±3.22	75.7 ±2.58	76.0 ±5.98	**78.7 ±5.48**	80.3	81.2	2.90*	−2.5
KernelDensity	76.8 ±4.49	75.7 ±3.95	73.1 ±5.13	**77.8 ±5.12**	77.0	79.0	1.00	−1.2
NaiveBayes	**79.9 ±4.56**	76.2 ±5.73	79.0 ±4.67	79.6 ±4.93	82.2	83.8	−.30	−4.2
OneR	65.0 ±4.94	65.3 ±4.85	76.9 ±4.18	**77.4 ±5.42**	70.6	70.6	12.1*	6.80*
PART	76.2 ±3.88	75.7 ±3.59	77.5 ±5.89	**79.0 ±6.24**	80.6	81.0	2.80*	−2.0
SMO	80.0 ±4.47	78.2 ±3.39	**82.3 ±4.27**	80.0 ±5.06	83.2	83.2	2.30	−.90

Table C.3: Results for dataset **N2F1** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	96.6 ±1.58	96.4 ±2.32	**97.1 ±1.37**	**97.1 ±1.66**	97.2	97.2	0.50	−.10
BoostingDT	96.5 ±1.78	96.3 ±2.06	**97.3 ±1.16**	96.9 ±1.45	97.9	96.6	0.80	−.60
DecisionStump	93.8 ±2.57	93.8 ±2.57	**96.3 ±1.64**	96.1 ±1.85	93.8	93.8	2.50	2.50*
DecisionTable	96.4 ±1.96	96.4 ±1.96	96.6 ±1.78	**96.9 ±1.85**	95.9	95.4	0.50	1.00
HyperPipes	90.3 ±2.91	89.9 ±3.38	**96.8 ±1.69**	93.5 ±3.24	90.8	89.4	6.50	6.00*
IB1	**97.1 ±1.37**	96.4 ±1.65	**97.1 ±1.20**	97.0 ±1.25	96.9	96.7	0.00	0.20
IB5	**97.7 ±1.49**	97.3 ±1.70	97.3 ±1.34	97.4 ±1.58	97.7	96.6	−.30	−.30
J48	96.8 ±1.62	96.2 ±1.55	96.6 ±1.65	**96.9 ±1.45**	96.6	95.3	0.10	0.30
KernelDensity	**97.2 ±1.55**	96.9 ±1.91	96.9 ±1.20	97.1 ±1.45	97.0	96.8	−.10	0.10
NaiveBayes	**97.7 ±0.95**	97.6 ±1.43	96.1 ±1.85	96.9 ±1.79	97.9	96.4	−.80	−1.0
OneR	93.3 ±2.75	93.3 ±2.75	95.4 ±2.41	**96.1 ±1.91**	93.3	93.3	2.80	2.80*
PART	96.3 ±1.83	95.9 ±2.18	96.1 ±1.29	**97.1 ±1.45**	95.2	96.0	0.80	1.10
SMO	96.8 ±1.03	96.9 ±1.20	**97.7 ±1.34**	97.4 ±1.58	97.3	96.1	0.80	0.40

Table C.4: Results for dataset **N2MN** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	88.1 ±3.11	**89.3 ±3.80**	88.1 ±2.13	88.1 ±4.18	88.2	87.7	−1.2	−.10
BoostingDT	**90.0 ±2.58**	89.0 ±5.73	88.7 ±3.59	89.5 ±3.63	87.7	87.4	−.50	1.80
DecisionStump	67.4 ±2.80	67.4 ±2.80	83.2 ±2.94	**83.5 ±3.84**	91.9	91.9	16.1*	−8.4*
DecisionTable	79.9 ±3.67	80.0 ±3.80	**84.3 ±3.13**	83.6 ±3.24	94.5	94.1	4.30*	−10*
HyperPipes	52.9 ±1.66	52.9 ±1.66	**63.8 ±3.01**	59.7 ±1.77	76.7	75.8	10.9*	−13*
IB1	82.7 ±3.37	80.7 ±4.72	**83.4 ±3.66**	82.9 ±3.45	94.0	93.8	0.70*	−11*
IB5	84.0 ±2.21	83.5 ±2.37	84.2 ±3.26	**84.3 ±2.83**	96.3	95.9	0.30*	−12*
J48	82.6 ±2.22	83.2 ±2.86	**85.6 ±5.34**	85.3 ±3.71	94.3	93.8	2.40*	−8.7*
KernelDensity	**84.5 ±3.17**	84.2 ±2.70	83.5 ±3.78	83.9 ±3.90	94.2	94.8	−.60*	−11*
NaiveBayes	80.6 ±2.95	80.0 ±3.30	**81.7 ±2.36**	81.6 ±2.59	95.1	95.0	1.10*	−13*
OneR	66.1 ±2.02	66.1 ±2.02	**83.1 ±3.51**	81.3 ±4.27	91.8	91.8	17.0*	−8.7*
PART	81.4 ±2.12	81.8 ±1.93	83.8 ±4.26	**84.3 ±3.65**	94.1	93.7	2.50*	−9.8*
SMO	84.7 ±3.86	84.7 ±3.86	**91.4 ±2.55**	84.8 ±3.71	95.9	95.7	6.70*	−4.5*

Table C.5: Results for dataset **N3F1** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	98.5 ±1.08	98.2 ±1.75	**98.7 ±0.67**	97.9 ±1.10	98.7	98.1	0.20	0.00
BoostingDT	**99.0 ±1.05**	98.5 ±2.07	98.4 ±1.07	97.8 ±1.14	98.7	98.2	−.60	−.30
DecisionStump	91.5 ±2.32	91.5 ±2.32	**98.2 ±0.92**	97.8 ±0.79	93.8	93.8	6.70*	4.40*
DecisionTable	97.0 ±1.56	97.1 ±1.60	**98.5 ±1.08**	97.8 ±1.55	97.8	97.0	1.40	0.70
HyperPipes	87.8 ±4.08	87.9 ±4.12	**98.2 ±1.03**	97.1 ±1.37	94.8	93.7	10.3*	3.40*
IB1	**98.3 ±1.64**	98.2 ±1.14	98.2 ±1.23	97.9 ±0.99	98.6	98.6	−.10	−.40
IB5	**98.6 ±1.07**	98.3 ±1.06	98.4 ±0.70	98.1 ±1.10	99.2	99.1	−.20	−.80*
J48	96.5 ±1.65	97.0 ±1.15	97.8 ±1.48	**98.1 ±1.10**	98.0	97.7	1.10	0.10
KernelDensity	98.3 ±1.64	**98.4 ±1.35**	98.1 ±1.10	**98.4 ±0.97**	98.7	98.8	0.00	−.40
NaiveBayes	98.0 ±1.05	98.1 ±1.10	**98.3 ±0.82**	97.6 ±0.70	99.1	99.2	0.20	−.90
OneR	90.8 ±3.36	90.8 ±3.36	**97.7 ±1.16**	**97.7 ±0.67**	92.7	92.9	6.90	4.80*
PART	97.4 ±1.58	96.5 ±1.18	98.4 ±1.07	**98.5 ±0.97**	98.3	98.0	1.10	0.20
SMO	**99.2 ±0.63**	98.4 ±0.84	99.1 ±0.57	98.5 ±0.71	99.5	99.1	−.10	−.40

Table C.6: Results for dataset **N3MN** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	84.8 ±3.58	83.7 ±3.27	**85.0 ±3.13**	84.1 ±1.37	84.3	84.5	0.20	0.50
BoostingDT	82.3 ±2.75	82.6 ±2.76	**83.5 ±4.06**	**83.5 ±3.31**	83.0	82.7	0.90	0.50
DecisionStump	72.1 ±4.53	72.1 ±4.53	83.1 ±3.78	**83.4 ±3.13**	76.2	76.2	11.3*	7.20*
DecisionTable	77.6 ±4.06	77.6 ±4.06	**83.7 ±3.06**	83.3 ±2.21	82.2	82.7	6.10*	1.00
HyperPipes	50.9 ±0.99	50.9 ±0.99	**72.9 ±4.12**	63.3 ±4.37	54.2	52.5	22.0*	18.7*
IB1	76.8 ±3.39	76.9 ±3.51	**79.9 ±3.60**	79.3 ±4.64	79.8	79.1	3.00	0.10
IB5	81.4 ±3.41	81.2 ±3.43	**83.7 ±3.74**	83.0 ±3.40	84.0	83.2	2.30	−.30
J48	77.9 ±3.90	78.0 ±3.83	79.7 ±4.40	**82.5 ±3.27**	81.7	82.2	4.50*	0.30
KernelDensity	77.1 ±3.75	77.0 ±3.62	**80.9 ±3.07**	79.5 ±5.02	79.9	80.8	3.80	0.10
NaiveBayes	80.9 ±4.75	80.9 ±4.75	**82.8 ±4.05**	82.4 ±3.63	85.4	85.8	1.90*	−3.0
OneR	62.6 ±6.64	62.6 ±6.64	81.2 ±3.26	**81.7 ±2.36**	75.5	75.5	19.1*	6.20*
PART	79.4 ±2.91	79.4 ±2.91	**82.7 ±3.77**	82.0 ±3.20	83.7	83.3	3.30*	−1.0
SMO	**87.1 ±2.77**	86.7 ±3.47	87.0 ±2.40	85.8 ±2.10	86.8	86.4	−.10	0.20

Table C.7: Results for dataset **N4F1** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	**92.9 ±2.64**	**92.9 ±2.18**	**92.9 ±2.64**	**92.9 ±2.18**	99.5	99.4	0.00*	−6.6*
BoostingDT	**95.8 ±2.66**	95.5 ±2.17	**95.8 ±2.66**	95.5 ±2.17	99.6	99.2	0.00*	−3.8*
DecisionStump	79.6 ±3.47	79.6 ±3.47	**96.9 ±2.13**	96.8 ±2.04	99.2	99.2	17.3*	−2.3*
DecisionTable	83.8 ±3.05	83.8 ±3.05	**98.6 ±1.07**	98.4 ±1.51	99.2	99.2	14.8*	−.60
HyperPipes	59.7 ±3.27	59.7 ±3.27	**98.1 ±1.45**	96.3 ±5.01	99.2	99.3	38.4*	−1.2*
IB1	95.7 ±3.20	95.6 ±2.72	96.7 ±1.70	**98.4 ±1.43**	98.5	99.5	2.70*	−1.1*
IB5	95.2 ±2.49	95.1 ±2.42	97.6 ±1.65	**98.6 ±1.35**	99.0	99.4	3.40*	−.80
J48	92.3 ±2.11	92.2 ±2.10	**99.1 ±0.88**	98.7 ±1.16	99.6	99.3	6.80*	−.50
KernelDensity	96.6 ±2.12	96.3 ±1.95	96.0 ±2.40	**98.4 ±1.51**	98.5	99.2	1.80*	−.80
NaiveBayes	84.8 ±3.29	85.0 ±3.23	95.8 ±2.57	**98.5 ±1.18**	99.8	99.8	13.5*	−1.3*
OneR	79.4 ±3.78	79.4 ±3.78	**97.0 ±2.11**	**97.0 ±2.11**	99.0	99.0	17.6*	−2.0*
PART	91.9 ±2.02	91.3 ±2.91	**98.9 ±1.20**	98.8 ±1.03	99.6	99.3	7.00*	−.70
SMO	97.4 ±1.90	97.1 ±1.97	**98.9 ±1.29**	97.7 ±1.34	99.2	98.5	1.50*	−.30

Table C.8: Results for dataset **N4MN** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	94.5 ±2.46	94.7 ±2.50	**99.1 ±0.99**	99.0 ±1.05	99.8	99.8	4.40*	−.70
BoostingDT	95.8 ±1.55	96.3 ±2.00	**99.3 ±0.82**	99.0 ±0.94	99.6	99.6	3.00*	−.30
DecisionStump	66.0 ±3.23	66.0 ±3.23	98.7 ±1.16	**98.8 ±1.14**	98.9	98.9	32.8*	−.10
DecisionTable	78.5 ±1.78	78.5 ±1.78	98.7 ±1.64	**98.8 ±1.14**	99.5	99.5	20.3*	−.70
HyperPipes	52.0 ±1.41	52.2 ±1.40	**99.3 ±1.06**	93.9 ±9.05	98.4	98.6	47.1*	0.70
IB1	97.6 ±1.65	97.8 ±1.48	**99.7 ±0.67**	99.2 ±0.79	99.8	100	1.90*	−.30
IB5	99.0 ±1.33	99.2 ±1.14	**99.5 ±0.85**	99.2 ±0.79	99.9	100	0.30*	−.50
J48	87.0 ±2.79	87.0 ±2.79	**99.3 ±0.82**	98.9 ±0.99	99.6	99.7	12.3*	−.40
KernelDensity	97.7 ±1.57	98.3 ±1.34	**99.6 ±0.84**	99.4 ±0.70	99.8	100	1.30*	−.40
NaiveBayes	81.7 ±3.92	81.8 ±3.94	**98.9 ±0.99**	**98.9 ±1.37**	99.8	99.9	17.1*	−1.0*
OneR	66.7 ±4.62	66.7 ±4.62	98.7 ±1.16	**98.8 ±1.14**	98.8	98.8	32.1*	0.00
PART	88.5 ±3.63	89.1 ±3.03	**99.1 ±1.10**	**99.1 ±0.74**	99.7	99.8	10.0*	−.70
SMO	**99.9 ±0.32**	**99.9 ±0.32**	99.8 ±0.42	99.1 ±0.99	99.9	100	−.10	−.20

Table C.9: Results for dataset **N5F1** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	90.1 ±4.07	89.7 ±4.00	**94.8 ±2.74**	94.2 ±3.49	96.3	96.0	4.70*	−1.5
BoostingDT	91.0 ±3.86	88.0 ±4.27	**95.7 ±1.95**	93.3 ±3.06	95.9	96.2	4.70*	−.50
DecisionStump	88.9 ±2.77	88.9 ±2.77	89.4 ±3.37	**89.8 ±2.97**	92.1	92.1	0.90*	−2.3
DecisionTable	88.1 ±2.42	88.5 ±2.37	93.7 ±3.02	**93.9 ±2.92**	96.3	96.2	5.40*	−2.4*
HyperPipes	60.8 ±3.05	61.1 ±2.47	**92.0 ±3.83**	80.3 ±7.89	85.8	86.2	30.9*	5.80*
IB1	83.6 ±2.22	84.3 ±3.92	90.2 ±3.77	**93.1 ±3.60**	94.0	95.1	8.80*	−2.0
IB5	89.2 ±2.53	89.4 ±3.92	91.9 ±3.60	**94.6 ±2.67**	95.2	96.6	5.20*	−2.0
J48	89.2 ±3.19	89.2 ±3.52	93.5 ±2.55	**94.3 ±2.36**	94.6	95.8	5.10*	−1.5
KernelDensity	83.7 ±2.26	87.2 ±3.19	87.7 ±3.59	**93.3 ±3.59**	94.1	95.4	6.10*	−2.1
NaiveBayes	89.6 ±2.80	89.5 ±3.92	89.5 ±3.60	**93.0 ±3.16**	96.5	96.9	3.40*	−3.9*
OneR	88.3 ±3.02	88.3 ±3.02	89.1 ±3.54	**90.0 ±3.53**	91.9	91.9	1.70*	−1.9
PART	88.1 ±5.09	88.3 ±3.16	93.4 ±3.20	**93.7 ±3.53**	95.5	95.8	5.40*	−2.1
SMO	94.5 ±3.06	90.6 ±4.84	**95.8 ±1.75**	94.9 ±3.00	96.6	96.4	1.30	−.80

Table C.10: Results for dataset **N5MN** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	90.2 ±4.52	91.7 ±3.59	**99.4 ±0.70**	**99.4 ±0.70**	99.9	99.9	7.70*	−.50
BoostingDT	94.2 ±2.49	95.0 ±3.43	**99.6 ±0.70**	99.4 ±0.70	99.9	99.9	4.60*	−.30
DecisionStump	70.8 ±3.68	68.5 ±3.75	**99.4 ±0.52**	99.3 ±0.67	99.8	99.8	28.6*	−.40
DecisionTable	77.9 ±6.35	73.5 ±8.53	**99.4 ±0.70**	**99.4 ±0.70**	99.9	99.9	21.5*	−.50
HyperPipes	51.1 ±2.13	50.9 ±1.91	**98.6 ±1.51**	90.0 ±12.7	99.9	84.6	47.5*	−1.3*
IB1	72.2 ±2.97	90.8 ±9.33	**99.5 ±0.97**	99.3 ±0.67	99.9	99.8	8.70*	−.40
IB5	90.7 ±3.06	93.9 ±5.59	**99.5 ±0.97**	99.4 ±0.52	100	99.9	5.60*	−.50
J48	83.6 ±4.50	88.6 ±3.60	**99.4 ±0.70**	**99.4 ±0.70**	99.9	99.9	10.8*	−.50
KernelDensity	72.6 ±2.84	91.8 ±8.32	96.6 ±1.43	**99.5 ±0.53**	99.9	99.9	7.70*	−.40
NaiveBayes	80.9 ±4.70	74.1 ±8.05	98.2 ±1.48	**99.6 ±0.52**	100	99.9	18.7*	−.40*
OneR	64.5 ±3.27	63.0 ±3.89	**99.4 ±0.52**	99.3 ±0.67	99.8	99.8	34.9*	−.40
PART	85.0 ±4.64	90.5 ±2.76	**99.4 ±0.70**	**99.4 ±0.70**	99.9	99.9	8.90*	−.50
SMO	**100 ±0.00**	98.4 ±2.27	99.8 ±0.42	99.6 ±0.52	100	99.9	−.20*	−.20

Table C.11: Results for dataset **N6F1** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	88.8 ±4.76	89.0 ±4.55	**95.4 ±2.59**	95.1 ±2.77	96.5	96.0	6.40*	−1.1
BoostingDT	89.2 ±4.61	88.1 ±4.68	**95.5 ±2.99**	94.6 ±2.67	96.3	96.2	6.30*	−.80
DecisionStump	81.0 ±5.14	81.0 ±5.14	88.8 ±5.37	**89.5 ±5.44**	92.1	92.1	8.50*	−2.6
DecisionTable	81.0 ±5.03	80.8 ±5.22	**94.6 ±3.06**	93.9 ±2.92	96.3	96.6	13.6*	−2.0
HyperPipes	52.3 ±1.25	52.0 ±1.25	**91.6 ±3.24**	82.9 ±7.14	85.2	86.1	39.3*	5.50*
IB1	85.1 ±2.47	83.5 ±4.77	91.8 ±2.39	**93.7 ±2.16**	92.9	94.6	8.60*	−.90
IB5	87.2 ±3.43	86.6 ±4.84	93.0 ±2.21	**95.6 ±1.90**	95.5	96.4	8.40*	−.80
J48	87.5 ±4.33	86.3 ±4.14	**93.4 ±3.27**	**93.4 ±2.59**	95.2	95.8	5.90*	−2.4
KernelDensity	85.4 ±2.46	84.9 ±3.60	87.1 ±3.84	**94.4 ±2.32**	92.9	95.0	9.00*	−.60
NaiveBayes	83.9 ±5.07	83.4 ±5.21	88.3 ±3.86	**94.1 ±2.92**	97.2	97.0	10.2*	−3.1*
OneR	79.3 ±4.42	79.3 ±4.42	**90.3 ±2.50**	88.8 ±5.47	91.9	91.9	11.0*	−1.6
PART	85.4 ±3.92	85.7 ±4.22	**95.0 ±3.16**	94.4 ±2.17	95.5	96.1	9.30*	−1.1
SMO	93.7 ±2.75	90.7 ±4.22	**95.9 ±2.08**	95.3 ±2.54	96.7	96.1	2.20*	−.80

Table C.12: Results for dataset **N6MN** with initial, GLOREF, and theoretical features.

Inducer	Initial att.		GLOREF		Theo.		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs Init.	vs Theo.
BaggingDT	87.5 ±1.72	87.9 ±1.79	**91.4 ±3.17**	89.0 ±3.89	91.6	89.3	3.50*	−.20
BoostingDT	88.1 ±3.35	88.0 ±2.98	**89.8 ±3.01**	89.0 ±2.54	91.0	87.6	1.70	−1.2
DecisionStump	67.3 ±4.69	67.3 ±4.69	**83.9 ±4.15**	**83.9 ±4.15**	79.3	79.3	16.6*	4.60*
DecisionTable	78.7 ±2.91	77.5 ±2.80	87.0 ±3.46	**87.5 ±3.72**	89.0	89.3	8.80*	−1.8
HyperPipes	53.1 ±2.08	53.3 ±1.89	**85.0 ±4.16**	70.8 ±8.18	67.7	67.3	31.7*	17.3*
IB1	70.8 ±3.49	76.1 ±6.05	**86.6 ±3.92**	86.4 ±4.40	85.2	86.5	10.5*	0.10
IB5	82.0 ±2.83	83.0 ±4.16	88.8 ±3.68	**89.8 ±2.86**	89.8	89.2	6.80*	0.00
J48	79.6 ±2.76	79.6 ±3.44	86.9 ±3.25	**88.0 ±2.75**	88.3	89.2	8.40*	−1.2
KernelDensity	71.0 ±3.40	76.6 ±5.46	85.8 ±4.02	**86.7 ±4.19**	85.2	86.8	10.1*	−.10
NaiveBayes	78.2 ±3.88	77.4 ±4.01	86.2 ±3.36	**88.0 ±3.53**	90.5	91.8	9.80*	−3.8*
OneR	63.9 ±5.45	63.9 ±5.45	82.9 ±2.56	**83.2 ±2.74**	79.4	79.4	19.3*	3.80
PART	82.1 ±4.86	80.0 ±3.71	**88.2 ±4.29**	87.6 ±2.59	89.6	88.9	6.10*	−1.4
SMO	91.9 ±3.07	89.9 ±3.96	**92.3 ±2.71**	90.8 ±3.16	92.4	92.3	0.40	−.10

C.2 Results with GLOREF on UCI datasets

Table C.13: Results for dataset **AUTO** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	**81.9 ±5.93**	78.7 ±9.87	80.0 ±7.21	74.7 ±9.35	−1.8
BoostingDT	**83.4 ±5.77**	83.1 ±9.95	82.0 ±5.34	80.1 ±8.37	−1.4
DecisionStump	45.0 ±5.17	36.6 ±8.41	**49.3 ±6.36**	43.5 ±9.39	4.38
DecisionTable	78.5 ±6.24	72.3 ±11.2	**80.0 ±5.75**	58.7 ±12.8	1.50
HyperPipes	55.0 ±10.5	28.9 ±5.79	**72.8 ±9.81**	39.5 ±12.1	17.7*
IB1	76.1 ±8.91	74.8 ±10.9	**79.5 ±7.53**	77.6 ±8.80	3.40
J48	**80.4 ±8.69**	70.9 ±11.9	77.2 ±11.2	71.3 ±9.14	−3.2
KernelDensity	**79.1 ±7.43**	68.0 ±13.8	1.4 ±2.30	73.3 ±9.08	−5.8
NaiveBayes	58.5 ±7.61	48.5 ±13.1	**61.0 ±8.08**	56.3 ±9.96	2.48
OneR	**63.5 ±11.4**	63.0 ±11.5	63.0 ±9.49	53.3 ±14.4	−.45
PART	73.7 ±6.35	71.2 ±11.5	76.1 ±7.32	**76.6 ±11.6**	2.93

Table C.14: Results for dataset **BALANCE-SCALE** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	82.2 ±3.98	82.2 ±3.98	**93.4 ±2.89**	92.3 ±4.43	11.2*
BoostingDT	79.2 ±4.57	79.2 ±4.57	**93.8 ±1.21**	92.8 ±3.80	14.6*
DecisionStump	56.8 ±2.84	56.8 ±2.84	**89.3 ±1.99**	89.0 ±1.93	32.5*
DecisionTable	74.9 ±7.31	74.9 ±7.31	90.2 ±1.41	**91.2 ±3.73**	16.3*
HyperPipes	46.1 ±0.59	46.1 ±0.59	**90.2 ±2.88**	83.2 ±4.94	44.2*
IB1	80.0 ±4.53	80.0 ±4.53	83.2 ±4.10	**87.2 ±5.69**	7.19*
J48	77.9 ±3.41	77.9 ±3.41	90.4 ±2.36	**90.9 ±3.75**	12.9*
KernelDensity	**88.5 ±2.68**	**88.5 ±2.68**	84.8 ±4.69	87.5 ±3.91	−.97
NaiveBayes	**90.7 ±1.22**	**90.7 ±1.22**	81.9 ±5.30	87.0 ±2.72	−3.7*
OneR	58.1 ±3.42	58.1 ±3.42	**90.6 ±2.78**	89.1 ±3.38	32.5*
PART	82.5 ±5.87	82.5 ±5.87	**91.7 ±2.87**	88.9 ±4.29	9.14*

Table C.15: Results for dataset **BREAST-W** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	96.4 ±1.69	95.3 ±2.61	**96.4 ±1.39**	95.9 ±1.71	0.00
BoostingDT	96.0 ±2.22	94.6 ±2.01	**96.9 ±1.76**	96.0 ±1.47	0.86
DecisionStump	92.4 ±2.79	91.4 ±2.53	**96.6 ±1.81**	95.6 ±2.17	4.15*
DecisionTable	95.3 ±2.26	94.6 ±1.88	94.9 ±2.15	**95.7 ±2.78**	0.44
HyperPipes	35.3 ±2.68	34.5 ±0.46	**96.6 ±1.93**	94.6 ±3.14	61.2*
IB1	**95.3 ±1.51**	94.6 ±2.92	95.0 ±1.82	94.4 ±2.91	−.29
J48	95.1 ±2.37	94.4 ±2.50	96.0 ±1.31	**96.0 ±2.59**	0.87
KernelDensity	95.1 ±1.54	**96.1 ±2.03**	46.3 ±5.57	95.1 ±2.63	−1.0
NaiveBayes	96.0 ±2.50	94.8 ±2.47	96.0 ±2.50	**96.6 ±1.93**	0.57
OneR	92.1 ±3.02	91.6 ±2.98	95.1 ±1.53	**95.7 ±2.23**	3.58*
PART	94.4 ±2.65	94.8 ±3.05	95.1 ±1.93	**95.7 ±2.77**	0.87

Table C.16: Results for dataset **CARS** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	**86.5 ±5.91**	85.7 ±4.02	81.4 ±6.26	82.7 ±8.75	−3.8
BoostingDT	84.4 ±7.14	**85.4 ±4.05**	81.1 ±7.59	81.6 ±7.35	−3.8
DecisionStump	66.4 ±7.13	66.4 ±7.13	**70.9 ±6.82**	70.2 ±8.14	4.57
DecisionTable	75.5 ±7.30	74.2 ±6.64	**77.6 ±5.23**	72.4 ±9.42	2.05
HyperPipes	63.1 ±7.99	62.3 ±7.55	**72.7 ±7.54**	64.8 ±5.01	9.67*
IB1	72.7 ±8.89	74.8 ±8.12	**80.9 ±7.28**	**80.9 ±5.60**	6.10
J48	**86.5 ±5.91**	83.7 ±3.15	79.1 ±8.43	80.9 ±7.07	−5.6
KernelDensity	74.0 ±6.86	74.8 ±4.37	**80.6 ±6.76**	80.1 ±6.83	5.83*
NaiveBayes	68.1 ±11.5	65.6 ±9.87	**69.7 ±12.0**	66.6 ±9.89	1.54
OneR	77.5 ±4.24	**77.8 ±4.10**	76.5 ±4.51	76.3 ±5.72	−1.3
PART	**81.9 ±9.56**	78.8 ±7.58	79.8 ±7.87	76.5 ±6.69	−2.0

Table C.17: Results for dataset **COLIC** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	85.1 ±7.11	**85.3 ±7.42**	82.9 ±5.95	80.2 ±6.02	−2.5
BoostingDT	79.1 ±6.66	**81.0 ±9.90**	80.2 ±4.35	79.4 ±7.36	−.79
DecisionStump	81.5 ±6.19	81.5 ±6.19	79.1 ±6.80	**82.4 ±6.07**	0.85
DecisionTable	82.1 ±7.50	**84.2 ±5.65**	80.7 ±6.50	80.2 ±7.14	−3.5
HyperPipes	41.9 ±5.06	42.3 ±2.97	**73.9 ±6.98**	. ±.	31.7*
IB1	77.4 ±6.21	**82.9 ±8.31**	81.0 ±5.36	78.3 ±4.02	−1.9
J48	85.1 ±6.61	**85.1 ±7.74**	78.3 ±4.49	80.7 ±4.91	−4.4
KernelDensity	79.1 ±4.13	**81.0 ±9.78**	76.6 ±4.71	78.6 ±5.65	−2.4
NaiveBayes	78.3 ±5.00	**83.2 ±6.54**	80.7 ±7.58	79.1 ±4.25	−2.5
OneR	**81.5 ±6.19**	81.5 ±6.19	78.6 ±6.83	81.3 ±6.35	−.23
PART	**84.5 ±8.03**	84.0 ±7.86	79.9 ±6.81	79.9 ±7.26	−4.6
SMO	82.4 ±7.20	**83.7 ±6.69**	80.5 ±8.49	82.9 ±8.21	−.80

Table C.18: Results for dataset **CREDIT-A** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	85.7 ±3.77	84.9 ±3.69	85.1 ±3.35	**85.9 ±2.37**	0.29
BoostingDT	83.9 ±3.64	**84.6 ±2.91**	84.5 ±4.16	83.8 ±4.47	−.14
DecisionStump	**85.5 ±3.62**	**85.5 ±3.62**	83.9 ±4.29	85.2 ±3.79	−.29
DecisionTable	83.8 ±2.54	**84.2 ±3.58**	83.5 ±4.98	83.0 ±4.83	−.72
HyperPipes	59.7 ±1.91	57.8 ±1.59	**76.8 ±3.62**	62.3 ±3.35	17.1*
IB1	**82.6 ±4.88**	73.3 ±12.6	**82.6 ±3.62**	81.9 ±6.16	0.00
J48	**84.9 ±4.39**	84.5 ±3.75	82.5 ±3.58	83.9 ±5.89	−1.0
KernelDensity	82.8 ±3.64	**84.5 ±2.90**	81.7 ±3.36	83.0 ±5.38	−1.4
NaiveBayes	77.5 ±3.63	80.6 ±5.21	84.9 ±4.17	**85.7 ±3.71**	5.07*
OneR	**85.5 ±3.62**	**85.5 ±3.62**	84.9 ±3.43	83.8 ±4.09	−.58
PART	83.8 ±3.79	83.3 ±3.94	81.6 ±5.59	**84.1 ±5.42**	0.29
SMO	84.5 ±3.55	85.2 ±3.73	**85.8 ±4.62**	85.4 ±4.01	0.58

Table C.19: Results for dataset **DIABETES** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	75.1 ±7.42	75.5 ±8.84	**76.4 ±6.59**	73.7 ±7.00	0.92
BoostingDT	71.6 ±8.51	72.0 ±7.93	**73.7 ±4.78**	73.2 ±7.51	1.70
DecisionStump	71.4 ±7.22	71.4 ±7.22	70.8 ±7.95	**76.2 ±4.97**	4.81
DecisionTable	**75.7 ±7.03**	74.7 ±7.52	74.2 ±6.00	75.1 ±5.70	−.53
HyperPipes	35.4 ±1.13	35.4 ±1.13	**65.8 ±6.37**	43.3 ±6.98	30.3*
IB1	69.0 ±6.52	66.4 ±5.45	70.4 ±6.20	**71.2 ±5.65**	2.21
J48	71.5 ±6.86	72.4 ±8.13	70.2 ±5.30	**74.5 ±5.73**	2.09
KernelDensity	**70.1 ±7.29**	69.7 ±7.21	63.3 ±5.55	69.1 ±5.76	−.91
NaiveBayes	76.3 ±8.23	75.3 ±7.61	75.9 ±8.31	**76.6 ±6.57**	0.25
OneR	72.3 ±6.73	72.1 ±6.79	**74.1 ±6.97**	73.4 ±5.17	1.83
PART	**74.4 ±6.24**	73.3 ±6.75	68.6 ±6.91	73.7 ±8.08	−.65
SMO	76.6 ±6.75	75.7 ±6.46	77.5 ±6.17	**77.7 ±5.63**	1.17

Table C.20: Results for dataset **GLASS** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	**74.3 ±12.1**	70.5 ±8.13	71.1 ±5.86	66.9 ±10.9	−3.2
BoostingDT	**75.2 ±8.31**	70.6 ±8.31	71.0 ±11.6	71.1 ±11.9	−4.1
DecisionStump	**44.9 ±2.39**	40.2 ±7.17	44.0 ±4.91	44.4 ±2.68	−.45
DecisionTable	**66.3 ±10.6**	64.3 ±11.9	62.7 ±11.4	59.8 ±12.1	−3.5
HyperPipes	45.9 ±8.08	31.4 ±12.3	**58.8 ±10.9**	31.7 ±10.2	12.9*
IB1	71.6 ±9.69	71.9 ±8.02	**73.9 ±10.8**	69.3 ±10.2	2.01
J48	66.8 ±12.4	**68.2 ±8.49**	66.9 ±13.7	60.3 ±11.5	−1.3
KernelDensity	69.2 ±6.38	62.5 ±12.7	**74.8 ±10.5**	70.1 ±9.51	5.63
NaiveBayes	48.6 ±9.03	38.2 ±11.6	**57.0 ±7.69**	53.2 ±11.4	8.40*
OneR	55.6 ±13.0	50.3 ±15.7	**60.7 ±13.9**	59.4 ±11.2	5.15
PART	**68.2 ±10.3**	68.2 ±9.49	68.2 ±8.44	66.9 ±10.5	−.02

Table C.21: Results for dataset **HEART** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	80.4 ±8.20	80.7 ±11.3	**83.3 ±7.46**	80.0 ±7.03	2.59
BoostingDT	77.8 ±11.3	73.0 ±9.08	**78.1 ±7.90**	77.4 ±7.70	0.37
DecisionStump	74.1 ±10.6	74.4 ±9.63	**79.6 ±5.86**	78.9 ±4.29	5.19
DecisionTable	**80.7 ±8.87**	80.4 ±6.99	79.3 ±7.24	78.1 ±6.86	−1.5
HyperPipes	45.6 ±3.05	44.8 ±2.10	**79.3 ±7.24**	56.3 ±9.21	33.7*
IB1	75.2 ±10.5	71.9 ±8.23	77.4 ±7.70	**78.9 ±6.06**	3.70
J48	**78.9 ±6.99**	76.7 ±6.54	78.1 ±7.29	74.4 ±7.29	−.74
KernelDensity	76.3 ±11.1	75.2 ±11.0	53.3 ±18.2	**82.6 ±4.64**	6.30
NaiveBayes	83.7 ±7.24	80.4 ±6.77	**85.2 ±8.19**	80.4 ±8.20	1.48
OneR	71.9 ±11.5	72.2 ±10.7	**80.7 ±5.74**	78.1 ±5.08	8.52*
PART	75.9 ±7.25	75.9 ±8.05	77.4 ±7.90	**80.4 ±8.01**	4.44
SMO	**84.8 ±8.81**	82.6 ±8.56	81.9 ±6.64	81.5 ±7.81	−3.0

Table C.22: Results for dataset **HEPATITIS** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	**84.5 ±8.75**	83.3 ±8.03	80.6 ±8.05	80.7 ±5.72	−3.8
BoostingDT	**82.1 ±8.03**	80.0 ±11.2	79.4 ±7.25	78.8 ±8.32	−2.7
DecisionStump	73.6 ±12.3	80.0 ±4.56	77.5 ±5.92	**81.3 ±8.34**	1.29
DecisionTable	76.8 ±8.42	82.0 ±7.73	83.2 ±6.97	**83.8 ±5.69**	1.87
HyperPipes	**83.9 ±4.28**	80.8 ±8.25	81.4 ±7.88	80.0 ±4.27	−2.5
IB1	81.4 ±9.96	73.5 ±11.6	**83.3 ±9.88**	78.6 ±8.32	1.92
J48	80.7 ±6.62	82.6 ±7.17	80.0 ±10.2	**85.9 ±6.53**	3.25
KernelDensity	80.1 ±10.6	**81.9 ±8.49**	80.9 ±12.0	81.2 ±6.60	−.67
NaiveBayes	**83.8 ±11.6**	81.9 ±10.6	80.2 ±10.6	82.7 ±10.5	−1.2
OneR	82.8 ±8.19	**83.3 ±7.21**	79.5 ±8.55	82.6 ±11.5	−.71
PART	80.1 ±8.55	82.0 ±8.92	79.4 ±10.7	**84.5 ±6.19**	2.58

Table C.23: Results for dataset **IONOSPHERE** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	91.7 ±5.47	87.2 ±4.51	**92.3 ±5.23**	88.3 ±4.29	0.58
BoostingDT	**93.4 ±4.48**	86.9 ±7.40	92.6 ±5.75	89.5 ±4.67	−.85
DecisionStump	82.3 ±6.31	76.3 ±6.67	**88.0 ±5.86**	84.1 ±5.03	5.70
DecisionTable	**91.2 ±5.47**	84.6 ±6.23	86.0 ±4.78	89.2 ±4.35	−2.0
HyperPipes	**92.9 ±4.91**	80.3 ±9.55	85.2 ±5.36	86.9 ±6.02	−6.0*
IB1	85.5 ±5.49	88.9 ±4.95	**90.3 ±5.26**	88.9 ±6.67	1.42
J48	**90.9 ±6.44**	88.9 ±4.38	90.0 ±3.64	90.6 ±4.28	−.27
KernelDensity	88.0 ±6.16	86.9 ±5.61	2.3 ±3.47	**88.6 ±4.04**	0.60
NaiveBayes	82.3 ±7.85	78.6 ±10.6	**92.3 ±6.88**	90.6 ±6.88	9.97*
OneR	82.6 ±5.98	81.2 ±6.80	**88.9 ±5.13**	87.2 ±4.47	6.29*
PART	88.0 ±6.72	87.2 ±3.40	88.9 ±5.46	**91.2 ±4.15**	3.14
SMO	87.7 ±5.07	77.2 ±7.86	**90.3 ±5.43**	86.3 ±6.85	2.56

Table C.24: Results for dataset **LIVER** with initial and GLOREF features.

Inducer	Initial attributes		GLOREF features		GLOREF
	All	FS	All	FS	vs Init.
BaggingDT	**71.0 ±8.29**	67.9 ±8.75	70.2 ±6.62	69.0 ±10.8	−.87
BoostingDT	**69.9 ±4.86**	64.1 ±9.44	68.1 ±6.76	65.2 ±6.04	−1.7
DecisionStump	59.7 ±9.83	59.0 ±10.6	67.6 ±8.32	**67.9 ±8.87**	8.15
DecisionTable	58.3 ±8.55	57.1 ±7.75	67.6 ±8.40	**68.2 ±8.40**	9.91*
HyperPipes	61.2 ±2.80	62.2 ±2.07	**68.7 ±4.55**	62.0 ±4.46	6.55*
IB1	62.4 ±7.35	59.2 ±10.9	64.1 ±6.40	**64.6 ±6.09**	2.29
J48	65.2 ±7.96	62.6 ±8.05	**67.6 ±8.30**	66.4 ±6.86	2.37
KernelDensity	64.1 ±5.41	61.8 ±8.12	65.2 ±5.80	**66.9 ±5.28**	2.84
NaiveBayes	55.1 ±7.99	53.6 ±8.57	64.1 ±6.13	**65.8 ±3.77**	10.7*
OneR	56.5 ±8.14	56.7 ±6.63	64.4 ±6.91	**65.6 ±8.00**	8.86*
PART	67.8 ±7.76	65.8 ±8.28	**71.0 ±11.3**	66.4 ±8.57	3.16
SMO	58.0 ±0.89	58.0 ±0.89	**74.2 ±7.44**	73.4 ±7.36	16.3*

C.3 Results for comparisons of GLOREF with PCA and ICA

Table C.25: Results for dataset **N1F1** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	95.4	**95.7**	95.4 ±2.37	95.5 ±2.22	94.8 ±3.22	94.4 ±3.13	0.20	0.90
BoostingDT	95.9	95.7	95.5 ±1.65	**96.0 ±1.56**	95.9 ±1.20	95.6 ±2.32	−.10	0.00
DecisionStump	**94.2**	94.0	87.1 ±3.81	87.1 ±3.81	89.6 ±5.58	89.6 ±5.58	7.10*	4.60*
DecisionTable	**95.9**	95.5	86.8 ±5.27	86.2 ±4.78	90.1 ±5.02	89.6 ±5.62	9.10*	5.80*
HyperPipes	**91.6**	79.0	57.7 ±2.95	55.1 ±3.25	59.4 ±2.55	57.1 ±2.56	33.9*	32.2*
IB1	95.5	**95.8**	95.3 ±2.45	**95.8 ±1.99**	95.1 ±2.51	95.1 ±2.33	0.00	0.70
IB5	95.5	96.1	95.8 ±1.81	**96.7 ±1.42**	95.7 ±2.00	96.1 ±2.38	−.60	0.00
J48	95.8	**95.9**	94.2 ±2.10	94.8 ±2.30	94.2 ±3.16	93.8 ±2.97	1.10	1.70
KernelDensity	95.6	95.9	95.3 ±2.45	**96.7 ±1.42**	95.2 ±2.49	95.7 ±2.21	−.80	0.20
NaiveBayes	93.0	**95.7**	88.9 ±4.68	88.8 ±3.99	90.0 ±4.67	90.0 ±4.97	6.80*	5.70*
OneR	94.0	**94.4**	87.1 ±4.91	87.1 ±4.91	88.8 ±5.39	88.8 ±5.39	7.30*	5.60*
PART	94.9	**95.4**	94.1 ±2.88	93.9 ±2.64	93.9 ±3.07	93.7 ±3.30	1.30	1.50
SMO	**97.5**	96.5	97.1 ±1.60	97.0 ±1.49	96.9 ±1.45	96.9 ±2.13	0.40	0.60

Table C.26: Results for dataset **N1MN** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	**79.5**	78.1	78.9 ±4.70	77.2 ±5.37	79.2 ±4.87	**79.5 ±5.62**	0.60	0.00
BoostingDT	78.1	**78.7**	77.6 ±4.65	76.2 ±4.61	77.1 ±4.07	**78.7 ±4.67**	1.10	0.00
DecisionStump	79.1	79.0	70.8 ±4.71	70.8 ±4.71	**80.0 ±5.14**	**80.0 ±5.14**	8.30*	−.90
DecisionTable	78.4	79.3	77.6 ±6.11	77.8 ±6.12	78.9 ±4.68	**80.5 ±4.70**	1.50	−1.2
HyperPipes	**63.2**	55.4	58.4 ±3.41	58.5 ±3.27	55.5 ±2.46	53.2 ±2.59	4.70*	7.70*
IB1	73.4	73.3	74.1 ±4.28	**75.3 ±4.27**	74.1 ±3.96	74.6 ±3.81	−1.9	−1.2
IB5	**78.6**	76.7	78.3 ±4.52	77.6 ±4.62	78.5 ±4.40	77.8 ±4.52	0.30	0.10
J48	76.0	78.7	77.4 ±4.43	78.0 ±3.94	**79.3 ±4.81**	**79.3 ±4.52**	0.70	−.60
KernelDensity	73.1	77.8	76.8 ±3.79	77.7 ±4.24	76.6 ±3.75	**79.8 ±4.71**	0.10	−2.0
NaiveBayes	79.0	79.6	**80.8 ±4.85**	78.6 ±6.40	80.6 ±4.65	80.2 ±5.61	−1.2	−1.0
OneR	76.9	**77.4**	66.7 ±6.73	66.7 ±6.73	75.4 ±5.99	75.4 ±5.99	10.7*	2.00
PART	77.5	79.0	76.5 ±6.47	77.7 ±5.46	**79.3 ±5.03**	**79.3 ±4.99**	1.30	−.30
SMO	**82.3**	80.0	81.1 ±5.43	80.2 ±5.05	81.2 ±5.22	80.4 ±4.84	1.20	1.10

Table C.27: Results for dataset **N2F1** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	97.1	97.1	96.6 ±1.90	96.8 ±1.87	**97.4 ±0.97**	**97.4 ±1.26**	0.30	−.30
BoostingDT	97.3	96.9	97.1 ±2.73	97.2 ±2.15	**97.4 ±1.35**	97.0 ±1.05	0.10	−.10
DecisionStump	96.3	96.1	92.2 ±2.82	92.6 ±3.20	**97.2 ±1.23**	**97.2 ±1.23**	3.70*	−.90
DecisionTable	96.6	96.9	96.1 ±1.66	95.8 ±2.97	**97.4 ±1.35**	**97.4 ±1.35**	0.80	−.50
HyperPipes	**96.8**	93.5	91.6 ±3.63	89.5 ±4.86	94.5 ±2.55	93.1 ±2.69	5.20*	2.30*
IB1	**97.1**	97.0	**97.1 ±1.10**	96.5 ±1.51	96.9 ±1.45	96.8 ±1.32	0.00	0.20
IB5	97.3	97.4	97.5 ±1.43	97.6 ±1.17	97.7 ±1.34	**97.9 ±1.20**	−.20	−.50
J48	96.6	96.9	96.5 ±2.32	96.3 ±2.63	96.8 ±1.48	**97.2 ±1.32**	0.40	−.30
KernelDensity	96.9	97.1	97.2 ±1.03	97.0 ±1.15	97.2 ±1.48	**97.8 ±1.32**	−.10	−.70
NaiveBayes	96.1	96.9	97.7 ±1.06	96.5 ±1.51	**97.8 ±1.32**	97.0 ±1.63	−.80	−.90
OneR	95.4	96.1	93.3 ±3.30	93.7 ±3.37	**97.1 ±1.10**	**97.1 ±1.10**	2.40	−1.0
PART	96.1	**97.1**	**97.1 ±1.73**	96.7 ±1.70	**97.1 ±1.29**	97.0 ±1.33	0.00	0.00
SMO	**97.7**	97.4	97.1 ±1.37	96.9 ±1.29	97.2 ±1.32	97.0 ±1.33	0.60	0.50

Table C.28: Results for dataset **N2MN** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	**88.1**	**88.1**	82.8 ±2.62	82.4 ±2.95	81.5 ±2.51	81.2 ±4.02	5.30*	6.60*
BoostingDT	88.7	**89.5**	83.7 ±2.75	82.3 ±3.33	82.5 ±3.81	80.8 ±3.99	5.80*	7.00*
DecisionStump	83.2	**83.5**	73.8 ±3.97	73.8 ±3.97	80.1 ±3.07	80.1 ±3.07	9.70*	3.40*
DecisionTable	**84.3**	83.6	79.9 ±3.87	80.1 ±3.78	81.2 ±2.66	81.1 ±2.81	4.20*	3.10*
HyperPipes	**63.8**	59.7	53.2 ±1.87	52.4 ±1.84	56.2 ±1.81	55.7 ±2.40	10.6*	7.60*
IB1	**83.4**	82.9	79.0 ±2.26	76.7 ±3.95	78.7 ±2.36	74.9 ±2.47	4.40*	4.70*
IB5	84.2	**84.3**	81.7 ±3.37	80.9 ±3.00	81.7 ±3.65	80.3 ±3.77	2.60	2.60
J48	**85.6**	85.3	81.1 ±4.07	81.2 ±3.91	79.4 ±3.06	79.2 ±3.01	4.40*	6.20*
KernelDensity	83.5	**83.9**	80.3 ±2.98	79.5 ±2.42	79.4 ±2.01	80.3 ±3.47	3.60*	3.60*
NaiveBayes	**81.7**	81.6	80.6 ±2.41	80.4 ±2.95	80.8 ±2.86	80.2 ±4.34	1.10	0.90
OneR	**83.1**	81.3	70.9 ±4.46	70.9 ±4.46	78.6 ±3.44	78.6 ±3.44	12.2*	4.50*
PART	83.8	**84.3**	80.6 ±4.33	79.7 ±4.37	79.5 ±3.60	79.4 ±3.34	3.70	4.80*
SMO	**91.4**	84.8	84.5 ±4.03	84.2 ±4.13	84.6 ±3.95	83.6 ±3.17	6.90*	6.80*

Table C.29: Results for dataset **N3F1** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	98.7	97.9	98.6 ±1.07	98.3 ±1.06	**99.0 ±0.94**	98.9 ±1.10	0.10	−.30
BoostingDT	98.4	97.8	98.6 ±1.17	97.9 ±0.99	**99.0 ±0.94**	98.8 ±1.03	−.20	−.60
DecisionStump	98.2	97.8	97.1 ±1.29	97.1 ±1.29	**98.8 ±1.03**	**98.8 ±1.03**	1.10*	−.60
DecisionTable	98.5	97.8	97.0 ±1.05	96.7 ±1.06	**98.8 ±1.03**	98.7 ±1.06	1.50*	−.30
HyperPipes	**98.2**	97.1	95.7 ±2.58	95.2 ±2.39	96.8 ±2.25	95.4 ±2.50	2.50*	1.40
IB1	98.2	97.9	**98.8 ±1.40**	98.0 ±1.33	**98.8 ±1.14**	98.4 ±1.17	−.60	−.60
IB5	98.4	98.1	98.9 ±0.99	98.3 ±1.06	**99.1 ±0.88**	98.9 ±0.88	−.50	−.70
J48	97.8	98.1	98.0 ±1.25	98.1 ±0.99	**98.8 ±1.03**	98.7 ±1.06	0.00	−.70
KernelDensity	98.1	98.4	**98.8 ±1.40**	98.4 ±1.26	**98.8 ±1.14**	**98.8 ±1.03**	−.40	−.40
NaiveBayes	98.3	97.6	97.9 ±1.37	96.9 ±2.28	98.8 ±0.63	**99.1 ±0.88**	0.40	−.80*
OneR	97.7	97.7	96.9 ±1.37	96.9 ±1.37	**99.1 ±0.74**	**99.1 ±0.74**	0.80	−1.4*
PART	98.4	98.5	98.0 ±1.41	97.7 ±1.42	**98.8 ±1.03**	98.7 ±1.06	0.50	−.30
SMO	99.1	98.5	**99.5 ±0.71**	98.6 ±1.35	**99.5 ±0.71**	99.2 ±0.92	−.40	−.40

Table C.30: Results for dataset **N3MN** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	85.0	84.1	83.9 ±4.01	84.5 ±3.69	**86.2 ±2.30**	86.0 ±2.40	0.50	−1.2
BoostingDT	83.5	83.5	83.9 ±3.73	82.7 ±4.69	**84.4 ±3.06**	83.6 ±2.84	−.40	−.90
DecisionStump	83.1	83.4	74.7 ±5.79	74.7 ±5.79	**85.5 ±4.01**	**85.5 ±4.01**	8.70*	−2.1
DecisionTable	83.7	83.3	82.7 ±2.79	82.9 ±3.60	**86.4 ±3.50**	85.7 ±4.27	0.80	−2.7
HyperPipes	**72.9**	63.3	54.8 ±1.48	54.3 ±2.00	57.8 ±3.01	58.0 ±2.83	18.1*	14.9*
IB1	79.9	79.3	77.2 ±4.10	78.2 ±4.02	77.4 ±3.69	**80.5 ±4.06**	1.70	−.60
IB5	83.7	83.0	82.6 ±3.06	82.4 ±3.41	82.3 ±2.87	**85.1 ±3.60**	1.10	−1.4
J48	79.7	82.5	81.3 ±3.33	79.3 ±4.57	**84.9 ±3.11**	84.5 ±3.34	1.20	−2.4
KernelDensity	80.9	79.5	77.1 ±4.31	78.3 ±4.00	77.3 ±3.56	**83.2 ±3.65**	2.60	−2.3
NaiveBayes	82.8	82.4	84.5 ±3.75	83.4 ±3.95	85.7 ±3.65	**86.2 ±3.33**	−1.7	−3.4
OneR	81.2	81.7	73.4 ±4.77	73.4 ±4.77	**84.2 ±3.39**	**84.2 ±3.39**	8.30*	−2.5
PART	82.7	82.0	80.5 ±5.23	81.6 ±3.92	**84.9 ±2.96**	84.8 ±2.86	1.10	−2.2
SMO	87.0	85.8	87.5 ±2.95	87.2 ±2.49	87.5 ±2.95	**87.6 ±2.32**	−.50	−.60

Table C.31: Results for dataset **N4F1** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	92.9	92.9	92.8 ±2.53	92.7 ±1.42	**94.1 ±1.73**	93.4 ±2.91	0.10	−1.2
BoostingDT	**95.8**	95.5	94.8 ±1.62	93.5 ±1.65	94.6 ±2.01	92.8 ±2.44	1.00	1.20
DecisionStump	**96.9**	96.8	79.6 ±3.47	79.6 ±3.47	93.2 ±2.57	93.2 ±2.57	17.3*	3.70*
DecisionTable	**98.6**	98.4	89.6 ±3.13	89.3 ±2.50	93.3 ±2.50	93.2 ±2.70	9.00*	5.30*
HyperPipes	**98.1**	96.3	61.3 ±3.23	59.8 ±2.44	71.3 ±3.33	68.8 ±4.33	36.8*	26.8*
IB1	96.7	**98.4**	94.3 ±3.02	92.6 ±3.81	94.4 ±2.95	91.3 ±3.62	4.10*	4.00*
IB5	97.6	**98.6**	94.4 ±1.90	94.0 ±3.13	94.4 ±1.71	93.9 ±3.31	4.20*	4.20*
J48	**99.1**	98.7	91.8 ±1.62	91.3 ±2.06	92.2 ±2.66	93.2 ±2.57	7.30*	5.90*
KernelDensity	96.0	**98.4**	94.6 ±2.95	93.9 ±3.18	94.7 ±2.67	94.2 ±2.70	3.80*	3.70*
NaiveBayes	95.8	**98.5**	89.5 ±3.34	88.9 ±4.04	90.8 ±2.15	92.8 ±2.78	9.00*	5.70*
OneR	**97.0**	**97.0**	79.4 ±3.78	79.4 ±3.78	92.6 ±3.03	92.6 ±3.03	17.6*	4.40*
PART	**98.9**	98.8	91.0 ±1.83	90.3 ±0.95	93.1 ±2.02	93.4 ±2.27	7.90*	5.50*
SMO	**98.9**	97.7	96.3 ±2.06	96.4 ±2.32	96.0 ±1.94	94.7 ±2.71	2.50*	2.90*

Table C.32: Results for dataset **N4MN** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	99.1	99.0	98.2 ±1.55	98.4 ±1.78	**99.9 ±0.32**	**99.9 ±0.32**	0.70	−.80*
BoostingDT	99.3	99.0	98.9 ±1.45	98.6 ±1.51	**99.9 ±0.32**	**99.9 ±0.32**	0.40	−.60*
DecisionStump	98.7	98.8	89.9 ±3.25	89.9 ±3.25	**99.9 ±0.32**	**99.9 ±0.32**	8.90*	−1.1*
DecisionTable	98.7	98.8	98.2 ±1.55	97.6 ±2.12	**99.9 ±0.32**	**99.9 ±0.32**	0.60	−1.1*
HyperPipes	99.3	93.9	78.1 ±3.11	77.1 ±2.42	99.2 ±0.79	**100 ±0.00**	21.2*	−.70
IB1	99.7	99.2	99.6 ±0.70	99.6 ±0.52	99.6 ±0.70	**99.8 ±0.42**	0.10	−.10
IB5	99.5	99.2	99.8 ±0.42	99.7 ±0.48	**99.9 ±0.32**	**99.9 ±0.32**	−.30	−.40
J48	99.3	98.9	98.1 ±1.37	98.2 ±1.40	**99.9 ±0.32**	**99.9 ±0.32**	1.10*	−.60*
KernelDensity	99.6	99.4	99.6 ±0.70	99.6 ±0.52	99.6 ±0.70	**99.9 ±0.32**	0.00	−.30
NaiveBayes	98.9	98.9	**99.9 ±0.32**	97.2 ±2.04	99.7 ±0.48	**99.9 ±0.32**	−1.0*	−1.0*
OneR	98.7	98.8	88.5 ±3.57	88.5 ±3.57	**99.9 ±0.32**	**99.9 ±0.32**	10.3*	−1.1*
PART	99.1	99.1	97.7 ±1.83	98.4 ±1.58	**99.9 ±0.32**	**99.9 ±0.32**	0.70	−.80*
SMO	99.8	99.1	**99.9 ±0.32**	**99.9 ±0.32**	**99.9 ±0.32**	**99.9 ±0.32**	−.10	−.10

Table C.33: Results for dataset **N5F1** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	94.8	94.2	93.1 ±2.51	92.5 ±2.92	**94.9 ±2.56**	94.5 ±2.72	1.70	−.10
BoostingDT	**95.7**	93.3	93.7 ±3.43	92.4 ±3.13	94.2 ±2.10	94.0 ±3.02	2.00	1.50
DecisionStump	89.4	89.8	88.9 ±2.77	88.9 ±2.77	**94.2 ±2.90**	**94.2 ±2.90**	0.90	−4.4*
DecisionTable	93.7	93.9	90.9 ±2.13	90.3 ±2.45	93.5 ±3.14	**94.3 ±3.09**	3.00*	−.40
HyperPipes	**92.0**	80.3	62.9 ±2.88	60.9 ±2.88	75.6 ±2.88	75.2 ±3.01	29.1*	16.4*
IB1	90.2	**93.1**	89.4 ±1.65	91.6 ±1.43	90.4 ±1.51	91.0 ±2.26	1.50	2.10
IB5	91.9	**94.6**	94.2 ±1.87	93.7 ±2.41	**94.6 ±1.58**	94.0 ±2.94	0.40	0.00
J48	93.5	**94.3**	91.3 ±2.31	91.3 ±2.71	93.7 ±2.58	93.5 ±2.84	3.00*	0.60
KernelDensity	87.7	**93.3**	89.3 ±1.57	91.6 ±1.17	90.4 ±1.78	91.6 ±2.32	1.70	1.70
NaiveBayes	89.5	93.0	92.5 ±3.17	92.0 ±2.91	**94.2 ±2.62**	**94.2 ±2.20**	0.50	−1.2
OneR	89.1	90.0	88.3 ±3.02	88.3 ±3.02	**93.9 ±3.14**	**93.9 ±3.14**	1.70	−3.9*
PART	93.4	**93.7**	90.1 ±3.00	91.5 ±2.37	93.5 ±2.37	**93.7 ±2.26**	2.20	0.00
SMO	95.8	94.9	**96.2 ±1.48**	94.8 ±3.26	**96.2 ±1.62**	94.9 ±2.18	−.40	−.40

Table C.34: Results for dataset **N5MN** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	**99.4**	**99.4**	92.3 ±3.09	92.3 ±2.45	95.4 ±1.17	95.3 ±1.34	7.10*	4.00*
BoostingDT	**99.6**	99.4	94.0 ±2.21	93.7 ±3.09	96.5 ±1.58	94.5 ±1.96	5.60*	3.10*
DecisionStump	**99.4**	99.3	70.5 ±3.50	70.5 ±3.50	94.8 ±2.25	94.8 ±2.25	28.9*	4.60*
DecisionTable	**99.4**	**99.4**	87.2 ±2.10	87.9 ±1.79	94.5 ±2.22	94.6 ±2.27	11.5*	4.80*
HyperPipes	**98.6**	90.0	54.1 ±2.13	53.6 ±1.65	77.4 ±2.88	77.6 ±2.79	44.5*	21.0*
IB1	**99.5**	99.3	80.6 ±3.44	89.4 ±4.93	88.5 ±2.12	94.5 ±3.27	10.1*	5.00*
IB5	**99.5**	99.4	92.6 ±1.78	93.8 ±2.39	93.9 ±1.91	95.3 ±2.41	5.70*	4.20*
J48	**99.4**	**99.4**	89.1 ±2.73	87.3 ±3.97	95.0 ±2.11	94.8 ±1.99	10.3*	4.40*
KernelDensity	96.6	**99.5**	80.7 ±3.56	89.5 ±4.97	86.7 ±2.45	95.8 ±2.15	10.0*	3.70*
NaiveBayes	98.2	**99.6**	91.3 ±3.53	90.6 ±2.84	94.5 ±1.84	94.6 ±1.78	8.30*	5.00*
OneR	**99.4**	99.3	70.4 ±4.09	70.4 ±4.09	95.1 ±1.91	95.1 ±1.91	29.0*	4.30*
PART	**99.4**	**99.4**	88.4 ±3.78	88.4 ±3.78	93.8 ±2.15	94.6 ±2.22	11.0*	4.80*
SMO	99.8	99.6	**100 ±0.00**	97.9 ±2.81	**100 ±0.00**	96.7 ±2.67	−.20	−.20

Table C.35: Results for dataset **N6F1** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	**95.4**	95.1	92.5 ±2.72	92.4 ±3.10	93.8 ±3.05	92.9 ±3.18	2.90*	1.60
BoostingDT	**95.5**	94.6	92.8 ±2.70	92.3 ±2.31	92.4 ±3.06	93.8 ±2.78	2.70*	1.70
DecisionStump	88.8	89.5	81.0 ±5.14	81.0 ±5.14	**92.7 ±2.91**	**92.7 ±2.91**	8.50*	−3.2
DecisionTable	**94.6**	93.9	91.0 ±2.21	90.7 ±2.67	92.1 ±3.21	92.6 ±2.76	3.60*	2.00
HyperPipes	**91.6**	82.9	53.4 ±1.78	53.1 ±2.28	65.0 ±3.06	64.7 ±3.06	38.2*	26.6*
IB1	91.8	**93.7**	88.7 ±2.75	90.5 ±2.01	88.9 ±2.73	89.7 ±1.70	3.20*	4.00*
IB5	93.0	**95.6**	91.0 ±2.75	92.3 ±2.54	91.5 ±2.17	92.2 ±1.93	3.30*	3.40*
J48	**93.4**	**93.4**	90.6 ±3.44	91.1 ±3.03	93.2 ±3.12	93.2 ±2.94	2.30	0.20
KernelDensity	87.1	**94.4**	88.9 ±2.69	90.8 ±2.30	89.2 ±2.62	89.9 ±1.37	3.60*	4.50*
NaiveBayes	88.3	**94.1**	89.8 ±3.85	88.7 ±3.56	91.7 ±3.20	93.2 ±3.29	4.30*	0.90
OneR	90.3	88.8	79.3 ±4.42	79.3 ±4.42	**92.2 ±2.94**	**92.2 ±2.94**	11.0*	−1.9
PART	**95.0**	94.4	89.0 ±2.49	89.5 ±3.84	93.3 ±3.59	92.2 ±3.43	5.50*	1.70
SMO	**95.9**	95.3	95.5 ±2.07	93.6 ±2.59	95.5 ±2.42	94.3 ±3.13	0.40	0.40

Table C.36: Results for dataset **N6MN** with GLOREF, PCA, and ICA features.

Classifier	GLOREF		PCA		ICA		GLOREF	GLOREF
	All	FS	All	FS	All	FS	vs PCA	vs ICA
BaggingDT	91.4	89.0	89.5 ±3.03	88.7 ±2.41	**91.7 ±2.87**	**91.7 ±3.20**	1.90	−.30
BoostingDT	89.8	89.0	88.4 ±4.12	88.7 ±3.23	90.1 ±3.00	**90.2 ±2.35**	1.10	−.40
DecisionStump	83.9	83.9	71.3 ±4.72	71.3 ±4.72	**91.9 ±2.88**	**91.9 ±2.88**	12.6*	−8.0*
DecisionTable	87.0	87.5	85.0 ±3.68	84.1 ±2.38	91.5 ±2.72	**91.9 ±2.88**	2.50	−4.4*
HyperPipes	**85.0**	70.8	58.7 ±2.58	55.5 ±1.27	80.3 ±4.06	80.1 ±3.55	26.3*	4.70*
IB1	86.6	86.4	77.2 ±3.01	81.7 ±4.74	82.4 ±3.60	**89.2 ±3.65**	4.90*	−2.6
IB5	88.8	89.8	86.4 ±4.35	86.6 ±4.09	88.8 ±4.08	**91.8 ±2.53**	3.20	−2.0
J48	86.9	88.0	84.6 ±3.69	85.3 ±3.16	90.1 ±3.14	**91.0 ±3.43**	2.70	−3.0*
KernelDensity	85.8	86.7	77.2 ±3.01	81.8 ±4.94	80.5 ±3.31	**92.2 ±3.71**	4.90*	−5.5*
NaiveBayes	86.2	88.0	89.2 ±3.43	84.3 ±3.13	90.5 ±3.21	**92.3 ±2.63**	−1.2	−4.3*
OneR	82.9	83.2	65.7 ±5.10	66.6 ±5.23	**91.9 ±3.78**	**91.9 ±3.78**	16.6*	−8.7*
PART	88.2	87.6	87.7 ±4.03	85.6 ±3.60	90.9 ±2.28	**91.3 ±2.95**	0.50	−3.1
SMO	92.3	90.8	**92.7 ±2.87**	91.8 ±2.66	92.6 ±3.10	92.6 ±2.88	−.40	−.30

Notations

$|A|$ The number of elements in the set A, page 50

$|S|^{(w)}$ The sum of weights of the instances in a dataset S, page 57

$\mathrm{Dom}(X)$ The domain of the attribute X, page 50

$\mathrm{GainRatio}(Y, T; S)$ The gain-ratio of a split on an attribute Y at threshold T of a dataset S, page 58

$\Gamma(\boldsymbol{X}, Y; \Omega)$ A GLOREF transformation involving the explanatory attribute(s) $\boldsymbol{X}$, the response attribute Y, and the set of parameters Ω, page 85

λ_1 A compatibility characteristic that indicates majority class in L, page 66

λ_2 A compatibility characteristic that indicates majority class in R, page 66

Ω The set of parameters involved in the definition of a GLOREF transformation, page 86

Ω^* The optimal set of parameters for a GLOREF transformation involving a given set of attributes, page 86

T^* The threshold T that maximizes the relevance of a cut point for a given attribute Y and a given dataset S, page 86

$\mathrm{U}(Y, T; S)$ The measure used to represent the usefulness of a cut point in the relevance graph, page 62

$\mathrm{val}_X(s)$ The value taken by attribute X in instance s, page 50

C The class attribute with domain $\{c_1, c_2, \ldots, c_k\}$, page 50

L The *cumulative* set which contains the instances falling on the *left* hand side of a split on a given attribute, page 57

R The *balance* set which contains the instances falling on the *right* hand side of a split on a given attribute, page 57

S A training dataset, page 50

s One of the instances in the training dataset S, page 50

S_i A non-empty subset of S included in the partition of S, page 55

X An attribute, may indicate an *explanatory* or a *response* attribute, page 50

Y An attribute, usually indicates a *response* attribute, page 50

Z A new feature generated by the GLOREF approach, page 85

Glossary

attribute A characteristic or a trait of an instance. An attribute can take any values in a given domain(real, integer, nominal). In supervised learning, one of the attribute is named the *class attribute* and all others are named the *input attributes*. The objective is to predict the value of the class attribute by using the values of the input attributes. The class attribute is a nominal attribute. Synonymous are: variable and parameter.

cut point A cut point separates a dataset S into two distinct subsets based on the values of a given continuous attribute Y and a threshold T. The subsets generated by a cut point are named L and R, where L contains all instances from S for which the value of Y is smaller than T and R contains the remaining instances. As another interpretation, if we suppose that the values of Y are ordered from left to right, then the subset L contains all instances on the *left* side of T while the subset R contains all instances on the *right* side of T.

explanatory attribute Attribute interactions happen when one or several *explanatory* attributes influence the values of a *response* attribute. Explanatory attributes can be continuous or nominal. The explanatory attributes are also named the *independent* attributes.

feature A characteristic or a trait of an instance constructed from the input attributes.

global relevance Indicates that the relevance is computed using all the training instances available.

instance An object defined by a number of attribute values and a class value. Also named an example or an observation. A dataset is composed of N instances.

local relevance Indicates that the relevance is computed using a subset of all the training instances available.

relevance or attribute relevance Designates the usefulness of a given attribute or feature in predicting the values of the class attribute. Potential measures to evaluate the relevance includes: the information gain[Quinlan, 1983], the gain ratio[Quinlan, 1986], and the χ^2[Kass, 1980; Mingers, 1989]. The relevance of continous attributes is often computed from the relevances of its cut points.

response attribute Attribute interactions happen when one or several *explanatory* attributes influence the values of a *response* attribute. In this work, response attributes are assumed to be continuous attributes. A synonym for response attribute is *dependent* attribute.

split Refers to the results of applying a cut point(i.e., the creation of two subsets).

testing data The set of instances used to evaluate the performance models learned through learning. Only the testing instances are used when evaluating the relevance of the features generated by the GLOREF approach. The testing dataset is never used during the learning and feature generation steps.

training data The set of instances used by the learning algorithm to build classification models or by the feature generation process to construct new features.

validation data An optional set of instances that is used during the learning process for intermediate evaluations or for global parameter tunning.