On Data Mining in Context: Cases, Fusion and Evaluation

Proefschrift
ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.mr. P.F. van der Heijden,
volgens besluit van het College voor Promoties
te verdedigen op dinsdag 19 Januari 2010
klokke 16:15 uur
door
Petrus Wilhelmus Henricus van der Putten
geboren te Eindhoven
in 1971

Samenstelling van de promotiecommisie

Prof. dr. J. N. Kok (promotor)	Universiteit Leiden
Prof. dr. C. Soares	Universiteit Porto, Portugal
Prof. dr. T. Bäck	Universiteit Leiden
Prof. dr. H. Blockeel	Universiteit Leiden & Katholieke Universiteit Leuven, België
dr. A. Knobbe	Universiteit Leiden

ISBN: 978-90-8891143-9

Contents

Chapter 1

Introduction

It is in our nature to see patterns in things. Humans and other intelligent organisms do it to learn from the environment. Scientists assess patterns not just to validate their theories, but also to discover new ones. Business users such as marketeers try to understand, predict and influence customer behavior. Doctors apply their experience with previous cases to diagnose patients and choose the most promising treatment. So it is not surprising that in the academic field concerned with creating artificial intelligence (AI) there is a keen interest in giving systems capabilities to learn from experience, rather than providing it with all the knowledge, rules and strategies it needs to solve a problem. Terms commonly used for this are data mining and knowledge discovery, using automated techniques to discover interesting, meaningful and actionable patterns hidden in data.

Even though the term only became trendy in academic research in the mid nineties, data mining or more generally the problem of how to learn from data has been a topic of interest for a long time. For example, at the dawn of the computing and AI field over 60 years ago McCulloch & Pitts (1943) introduced neural networks that mimic how the brain learns, and the empirical revolution in science around four hundred years ago led to an increased interest in developing scientific methods to derive natural laws and theory from empirical observations. However, up to until only ten years ago, data mining had hardly left the research labs. Today, most people get exposed to data mining a couple times a day without even knowing: when Googling for a web site, looking at recommendations for books or CDs at Amazon.com or tuning into their TiVo digital video recorder. And within certain business areas, such as marketing or risk management, data mining is now common practice for business end users, not IT.

The themes and topics of this thesis can be explained through the title: 'On Data Mining in Context: Cases, Fusion and Evaluation'. The word 'context' is used here with two different angles in mind. Firstly, the word context indicates that the

research presented is motivated by practical applications, mostly from either business or biomedical domains. This is not to say that we focus on case applications only. We do aim to develop methodology and algorithms that are generalizable over a number of problem domains, but our research is driven by the problems and needs of data mining in practice.

Secondly, the word 'context' refers to the process of data mining and knowledge discovery. We feel that quite a large proportion of academic research effort in data mining is targeted at the core modeling step in the process, for example by extending existing or developing new algorithms for prediction, clustering or finding association rules. Whilst this is valuable research, we aim to focus more on developing methodology for supporting the steps preceding or following the core modeling step, such as objective formulation, data preparation, model & results evaluation and post-processing & deployment; or focus on the end to end process as a whole.

Without further qualification, this may sound as quite an ambitious research area for a single thesis. However it should be seen as an overarching research theme and objective, rather than a single research question. To keep this practical and meaningful, we will investigate and discuss a selection of specific topics that fit into the overall theme. In most cases the approach is to explore and introduce hopefully new ways to look at these problems, identify interesting areas for research and provide proof of concept examples, rather than producing technically detailed solutions. Hence this thesis will not contain extensive elaborations and extensions of algorithms and proofs. That said, barring some illustrative introductory cases in the second chapter, we aim to go beyond merely applying an existing algorithm or approach to a single practical problem. We realize that this results in a thesis that is neither completely business and application focused nor research and algorithm oriented, and that the discussion of topics will be broad rather than deep. Our objective is to purposely be on the border of applications and algorithms to contribute to bridging the gap between data mining practice and research, enabling a more widespread application of data mining.

1.1 Thesis Theme, Topics and Structure

Let us discuss the topics of this thesis in more detail, using the knowledge discovery and data mining process as the underlying structure. A generally accepted definition of data mining is:

> "The non-trivial process of identifying valid, novel, potentially useful and ultimately understandable patterns in data" (Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy 1996), p. 6.

Note that according to this definition data mining is a process. In the standard, classical view a number of steps are identified (see figure 1.1). First the problem

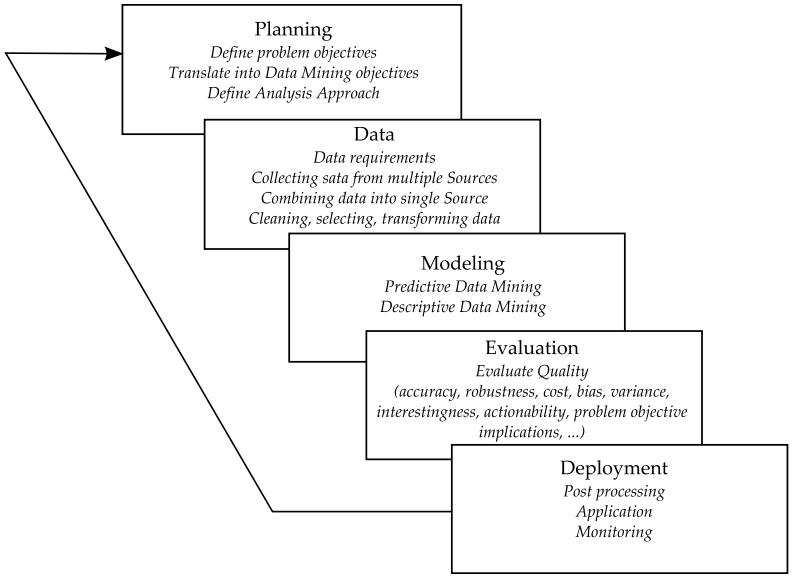


Figure 1.1: The Data Mining Process

needs to be defined in terms of business or scientific goals, and translated into specific data mining objectives and an analysis approach. The second step is the data step, sourcing raw data, combining and transforming it so that it can be used for the data mining task at hand. This is typically the most time consuming step, unless the process has been completely automated. The third step is the modeling step, algorithms are used to extract the actual patterns from the data, for predictive or descriptive data mining. In the fourth step these patterns and models are evaluated in terms of quality and content. In the final deployment step, the abstracted models are applied to new data, and the resulting output is combined with other information to take appropriate action (Chapman, Clinton, Khabaza, Reinartz & Wirth 1999), (Fayyad et al. 1996).

This standard view of the data mining process has been designed with a relatively traditional use case in mind, of a data mining expert who carries out a one off project to build a predictive model or generate useful descriptive patterns. One may for instance argue that this model doesn't really cover how to embed data mining in an organization, it doesn't address how to create a model or mining factory where a model library is continuously extended by data miners, coverage of the deployment steps is weak (i.e. the part of the lifecycle when the resulting models are actually used) nor does it really seem to fit fully automated, real time learning systems (see van der Putten (1999*b*), van der Putten (2002*c*) and van der Putten (2009) for some

Process Step	Chapter
Planning (and End to End)	Chapter 2: Motivating Examples
Data	Chapter 3: Data Fusion: More Data to Mine in
Modeling	Out of scope for this thesis
Evaluation	Chapter 4: Bias Variance Analysis of Real World Learning Chapter 5: Profiling Novel Algorithms
Deployment	Out of scope for this thesis

Table 1.1: Mapping thesis chapters against the data mining process.

non-academic papers addressing these topics). In this thesis however we will adopt the standard view as it is widely accepted and generally well known, and it is fit for purpose for organizing the thesis chapters.

To reiterate, the theme of this thesis is data mining in context. The context refers to the importance of the steps other than the core modeling step, the relevance of the end to end data mining process and the aim of developing methodologies and algorithms that are driven by data mining in practice without sacrificing general applicability across problems. The data mining process itself is thus used to organize the thesis chapters (see table 1.1).

The objective of *chapter 2* is to present selected end to end data mining cases that will serve as motivating examples for the importance of studying data mining in context, and identify high level lessons learned and areas for further research. The remaining chapters in the thesis focus more on specific process steps, research topics and solutions addressing some of these lessons learned.

The first case in chapter 2 is based on an early paper with examples of using descriptive and predictive data mining for direct marketing (see the next section for a full mapping of chapters against publications). This includes an insurance response modeling case and a review of a number of direct marketing projects from the early days of commercial data mining. The projects were carried out in the mid nineties, but the lessons from these cases are still valid today.

The second case is a similar example of introducing data mining to an end user audience with no data mining or computer science background. The goal in this case is to predict five year survival probability for head and neck cancer patients. So called evidence based medicine is becoming more important in the medical field, from empirically based studies towards medical decision support systems. We present some explorative predictive modeling results. The performance of the top classifiers is relatively close, and we carry out a specific analysis to get a better picture of what is causing any differences in performance.

The third case is concerned with the classification of yeast cells to evaluate pathogen conditions. This case takes a holistic view by showing the full end to end process from growing yeast samples, capturing images, feature extraction, super-

vised and unsupervised data mining and evaluation of the results. For this problem we demonstrate that all classifiers perform roughly the same; almost perfect performance in this case. In practice it can occur quite frequently that problems are either trivial or too hard to solve, whereas in typical machine learning papers problems are carefully selected to be 'interesting': not too hard nor too easy. That said, it is still an open question whether the underlying problem is easy to solve (classifying yeasts) given that the data mining problem is easy (classifying pictures). In our opinion this is a good example that in practice the translation of the research or business problem into a data mining problem and approach has a major impact on the results.

The fourth case introduces a real time automatic scene classifier for content based video retrieval in television archives. In our envisioned approach end users like archive documentalists, not image processing experts, build classifiers interactively, by simply indicating positive examples of a scene. A scene defines against which background a certain action takes place (day or night, city or countryside, inside or outside etc.). To produce classifiers that are sufficiently reliable we have developed a procedure for generating problem specific data preprocessors that extract rich, local semantic features relevant to the specific global settings to be recognized, exploiting end user knowledge of the world that identifies what building blocks may be useful to classify the scene . This approach has been successfully applied to a variety of domains of video content analysis, such as content based video retrieval in television archives, automated sewer inspection, and porn filtering. In our opinion in most circumstances the ideal approach would be to let end users create classifiers, because it will be more scalable – a lot more classifiers can be created in much shorter time, and it may eventually lead to higher quality classifiers compared to purely data driven approaches.

Chapter 3 is concerned with the data step in the data mining process. More specifically we introduce the topic of data fusion, which is not widely studied in data mining. A common assumption in data mining is that there is a single source data set to mine in. In practice however, information may be coming from different sources.

Take for instance the marketing domain. The wide majority of data mining algorithms require a single denormalized table as input, with one row per customer (examples of exceptions are multi relational data mining and semi structured data mining techniques). However, for a single customer information may be available from a variety of sources, for instance operational data systems, analytical data marts, survey data and competitive information. Linking information together about this customer can be seen as a simple join problem, or if common keys are missing a so called record linkage or exact matching problem. In our research however we focus on the situation when information about different customers (or other entities) is combined, the so called statistical matching problem. A typical example would be merging the information from a market survey among 10.000 customers with a customer database containing 10 million customers, by predicting the answers to the

survey for each customer in the database. This then results in a single customer table that can be used as a rich data source for various further data mining exercises.

We introduce the problem in a data mining and database marketing context and provide an example that demonstrates that it is indeed possible that data fusion can improve data mining results, by providing a richer, combined data set to mine in. However we also discuss some of the limitations of data fusion. In addition we provide a process model for fusion, as a main blueprint for designing a so called Data Fusion Factory, for fusing data sets following a standardized, industrialized procedure.

As outlined we focus on steps in the data mining process around the core modeling step, so *chapter 4* is mainly concerned with evaluation, not just of modeling but of the end to end process. We conducted a field experiment by providing data for a data mining competition. The CoIL Challenge 2000 attracted a wide variety of solutions, both in terms of approaches and performance. The goal of the competition was to predict who would be interested in buying a specific insurance product and to explain why people would buy. We had selected a problem representative for real world learning problems (as opposed to many standard machine learning benchmarks in our view). For instance it was important to align the data mining approach and evaluation with the business objective to get good results (scoring rather then classification), the data used was a combination of a few strong predictors and many irrelevant ones and to make matters worse we made it tempting to overfit the problem by offering a substantial prize.

Unlike most other competitions, the majority of participants provided a report describing the path to their solution. We use the framework of bias-variance decomposition of error to analyze what caused the wide range in prediction performance. We characterize the challenge problem to make it comparable to other problems and evaluate why certain methods work or not. We also include an evaluation of the submitted explanations by a marketing expert. We find that variance is the key component of error for this problem. Participants use various strategies in data preparation and model development that reduce variance error, such as attribute selection and the use of simple, robust and low variance learners like Naive Bayes. Adding constructed attributes, modeling with complex, weak bias learners and extensive fine tuning by the participants often increase the variance error.

In *chapter 5* a novel algorithm for classification is presented, however the topic of the chapter is actually model evaluation and profiling. We discuss an approach for benchmarking and profiling novel classification algorithms. We apply it to AIRS, an Artificial Immune System algorithm inspired by how the natural immune system recognizes and remembers intruders. We provide basic benchmarking results for AIRS, at the date of publication to our knowledge the first such test under standardized conditions. We then continue by outlining a best practice approach for 'profiling' a novel classifier beyond basic benchmarking.

The rationale behind this is as follows. From the No Free Lunch theorem we can conclude, loosely stated, that there is no classification algorithm that will perform better than all other algorithms across all problem domains. Of course it is possible to create an algorithm that performs consistently worse, and the main purpose of basic benchmarking is to provide a baseline test result to rule this worst case out. Taking the theorem into account, it can be concluded it is not very useful to use basic benchmarking to prove that some new algorithm is performing better than the rest; which often is the case in papers introducing a new algorithms. We claim it will be more relevant to identify when best to apply the novel algorithm and when not, for instance by relating problem domain properties such as data set size to relative performance patterns. Another approach to profiling novel algorithms is to empirically measure the similarity in behavior of the algorithm compared to others. We present three methods for computing algorithm similarity and find that AIRS compares to other learners that are similar from a theoretical point of view, but its behavior also corresponds to some specific other classification methods, which was a surprising result.

1.2 Publications

All chapters are largely based on previously published materials, which in some cases have been extended or combined for the purpose of the thesis. Below we list the specific publications for each chapter:

- Chapter 2: Motivating Examples
  - The review of various direct marketing data mining projects appeared as a chapter in a book on Complexity and Management (van der Putten 1999*a*). See also van der Putten (2002*a*) and van der Putten (2002*b*) for more extensive discussions of some of the provided examples. In addition we refer to some related academic and managerial publications in this section, among others van der Putten (1999*b*), van der Putten (1999*c*), van der Putten (1999*d*), van der Putten (2002*c*), van der Putten, Koudijs & Walker (2004), van der Putten, Koudijs & Walker (2006), van der Putten (2009).
  - The cancer survival classification case was published as an invited chapter in a book on Head and Neck Cancer targeted at medical professionals (van der Putten & Kok 2005).
  - The yeast classification case was presented at the ICPR and SPIE conferences (Liu, van der Putten, Hagen, Chen, Boekhout & Verbeek 2006), (van der Putten, Bertens, Liu, Hagen, Boekhout & Verbeek 2007).

  - We introduced the scene classification case in a BNAIC demo paper and a KDD workshop paper (Israel, van den Broek, van der Putten & den Uyl 2004*a*), (Israël, van den Broek, van der Putten & den Uyl 2004*b*), and provided a more extensive description in an invited chapter in a handbook on Multimedia Data Mining (Israël, van den Broek, van der Putten & den Uyl 2006).

- Chapter 3: Data Fusion: More Data to Mine in

  - This chapter is based on a number of conference and workshop papers, including a SIAM International Conference on Data Mining paper (van der Putten 2000*a*), (van der Putten 2000*b*), (van der Putten, Kok & Gupta 2002*b*). An earlier version of the SIAM paper was also published as a MIT Sloan School of Management Working Paper (van der Putten, Kok & Gupta 2002*a*). A paper on the process model appeared at the BNAIC conference (van der Putten, Ramaekers, den Uyl & Kok 2002). In revised format, the chapter has been accepted for a book on intelligent systems and soft computing for marketing, to be published in 2010.

- Chapter 4: Bias Variance Analysis of Real World Learning

  - This chapter is based on two collections of competition reports (van der Putten & van Someren 1999), (van der Putten & van Someren 2000) and a paper in the Machine Learning journal (van der Putten & van Someren 2004).

- Chapter 5: Profiling Novel Algorithms

  - This chapter is based on a number of conference papers (van der Putten & Meng 2005), (Meng, van der Putten & Wang 2005), (van der Putten, Meng & Kok 2008) along with selected previously unpublished materials.

Chapter 2

Motivating Examples

The overarching theme of this thesis is data mining in context. This refers to the importance of the steps around the core modeling step and the end to end data mining process as a whole. It also refers to the idea that we aim to develop methodologies and algorithms that are applicable and generalizable over a number of problem domains, but the research is also driven by the problems and needs of data mining in practice. In this chapter we will describe some data mining cases that will serve as motivating examples for the importance of studying data mining within this particular context. The remaining chapters in the thesis focus more on particular research topics and solutions. Each of the cases will preceded with a short section relating the case to the thesis.

2.1 Data Mining in Direct Marketing Databases

In direct marketing large amounts of customer data are collected that might have some complex relation to customer behavior. Data mining techniques can offer insight in these relations. In this case we give a basic introduction in the application of data mining to direct marketing. Best practices for data selection, algorithm selection and evaluation of results are described and illustrated with a number of real world examples. We suggest two lines of research that we consider important to put data mining in the hands of the marketeer: automating data mining techniques and integration of data mining in an open knowledge management framework (van der Putten 1999*a*), (van der Putten 2002*b*).

2.1.1 Introduction

In marketing, there are two opposed approaches to communication: mass media marketing and direct marketing. In mass media marketing, a single communication message is broadcast to all potential customers through media such as newspapers, magazines, outdoor communication, radio or television. Such an approach typically implies a high waste: only a small proportion of the customers communicated to will actually be interested in buying the product. Now that competition increases and markets get more fragmented the problem of waste worsens. Moreover, in spite of huge investments in market research and media planning, it is still hard to really quantify the benefits of mass media marketing. At best indications can be given how many people of what type were reached, but data on customer response is typically lacking.

These developments have led to an increased popularity of direct marketing, especially in the sectors of finance, insurance and telecommunication. The ultimate goal of direct marketing is cost-effective, two-way, one-to-one communication with individual customers. This is not limited to the web, the majority of direct marketing communication is still handled by traditional channels such as direct mail, email, sms and inbound and outbound calls. For effective direct marketing it is essential to learn present and predict future customer preferences. In today's business environment, customer preferences change dynamically and are too complex to derive straightforwardly.

Data mining, the continuous analysis of customer behavior patterns, may offer a flexible solution to this problem (Ling & Li 1998), (Berry & Linoff 1997). In this case description we will give a practical introduction to data mining for direct marketing purposes. We will not discuss any theoretical algorithmic issues, nor will we describe experiments in detail. We only aim to offer a managerial, self contained, tutorial style introduction to current data mining best practices for direct marketing: how is data mining commonly applied and evaluated, and which data and algorithms are most appropriate, given common direct marketing tasks.

In the first part we will describe the data mining process in a direct marketing context. A case from insurance is added to give an impression of the practical issues related to data mining projects, including the evaluation of data mining results. In the second part we will focus on lessons learned with respect to the selection of data and algorithms, based on eight data mining projects carried out in co-operation with the Dutch association for direct marketing, sales promotion and distance selling (DMSA) (Wagenaar 1997). We conclude with suggesting directions for research.

2.1.2 Data Mining Process and Tasks in Direct Marketing

Data mining can be defined as the extraction of valuable patterns that are hidden in large amounts of customer data (Fayyad et al. 1996). The end to end process of

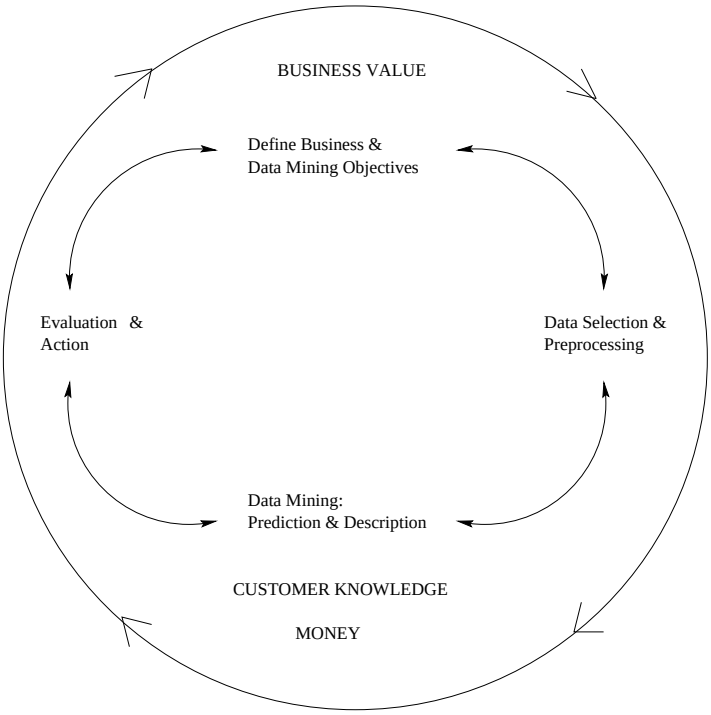


Figure 2.1: The Knowledge Discovery Cycle.

steps involved in data mining is sometimes referred to as the knowledge discovery cycle (see figure 2.1 and also section 1.1). This includes definition of the objectives, selection and preparation of the data and evaluation of the results with technical and business criteria.

Within the loop of a single project, it is not uncommon to go through the knowledge discovery cycle a number of times. For instance, by doing data mining analysis one might discover that some important data was not selected or was not prepared in the appropriate format. By performing different data mining projects repeatedly, an organization starts to learn more and more about customers, contributing to the 'institutional memory' of an organization. This can be considered to be a second loop of learning. Note however that this knowledge is usually not codified, integrated and disseminated in a systematic way; data mining and knowledge management tools and technologies supporting institutional learning are typically lacking.

The current success of data mining in businesses is enabled by a number of technical factors. Growing amounts of customer data are collected and made accessible

in corporate data warehouses, especially in industries where detailed tracking of customer behavior is required for operations or billing anyway. A telecommunications provider needs to charge its customers for calls and a bank needs carry out all the transactions that customers request. Powerful new data analysis algorithms are discovered by researchers from statistical pattern recognition and artificial intelligence fields such as machine learning, neural networks and evolutionary computation. Today, ordinary office computers are powerful enough to run these advanced data mining algorithms.

In a direct marketing context, two prototypical data mining objectives can be distinguished: *prediction* and *description*, see sections 2.1.3 and 2.1.4. Prediction involves predicting unknown or future customer behavior from known customer attributes. Description aims at discovering human interpretable patterns in the data. Best practice application of prediction and description in direct marketing are given below. For a detailed real world example we refer to the insurance case described in section 2.1.5. More managerial discussions of the data mining process can be found in van der Putten (1999*b*), van der Putten (2002*c*) and van der Putten (2009).

2.1.3 Prediction

The classical case in direct marketing for prediction is response modeling. Usually, the relative number of customers that responds to untargeted outbound direct mail, sms or email campaigns is very low (5% or less). Predictive models can be built to identify the prospects most likely to respond. Historical data about previous mailings or proxies such as natural product uptake are used to construct the model. If such information is unavailable, for instance when selecting prospects for a new product, a test campaign is performed to collect information for a small random sample from the relevant population in scope. The resulting model can be applied to filter prospects from the existing customer base or from external address lists acquired from commercial list brokers.

Although response analysis is by far the most common type of predictive modeling for direct marketing, other applications are promising as well, such as basic product propensity and usage modeling (see van der Putten (1999*c*), van der Putten (1999*d*)) for a credit card example), customer retention or estimating customer potential lifetime value (Paauwe, van der Putten & van Wezel 2007), and especially for the financial services industry, blending marketing decisions with credit risk decisions (van der Putten et al. 2004), (van der Putten et al. 2006).

2.1.4 Description

A shortcoming of prediction is that it produces models that, to a smaller or larger extent, may be perceived as black boxes. A response prediction model is useful

for selecting interesting prospects. But it does not necessarily give the marketeer more insight into the reasons why these customers respond. Several descriptive data mining tasks exist that may help out in this case. The main ones we will discuss here are profiling and segmentation.

Profiling is typically used to answer the question what the discriminating profile of a specific target group is. A particular approach is to discover which attribute-value pairs differ significantly between a selection and the database as a whole, or other reference groups. For example, we considered profiling useful for mining national media surveys in Holland and Belgium. Every questionnaire contained hundreds of questions on media interests, product consumption and socio-demographics, so it was infeasible to construct the profile of deviating attribute values manually. For example, by using profiling for an analysis of vodka drinkers we found that they are more often students, drink more Bacardi rum and are more frequent visitors of cinemas, compared to reference customers (van der Putten 2002*a*). The same technique can be used to mine customer databases rather than media surveys as we will demonstrate in the insurance case below.

In segmentation, the goal is to discover subgroups in data. Customers within a segment should resemble each other as much as possible, where as the segments should differ as much as possible. For example, in the vodka case we found out that the average vodka drinker does not really exist. Instead, subgroups were found that could be described as ”cocktail drinking teenagers”, ”young student couples” and ”traveling salesmen”. Various approaches to segmentation exist, the main ones are clustering and projection. In clustering the algorithm partitions the customers into a finite number of groups itself, in projection high dimensional data about customers is projected into two or three dimensions, and the user can interactive explore and label groups of customers in the lower dimensional space (van der Putten 2002*a*).

2.1.5 Insurance Case

We will illustrate the end to end process and the concepts of predictive and descriptive data mining with a direct marketing case from insurance. The business objective in this example was to expand the market for an existing consumer product, a caravan insurance, with only moderate cost investment. We identified two data mining objectives: selecting individual prospects and describing existing customers.

Data Selection and Preprocessing

Each customer was characterized by a selection of 85 input attributes plus a target attribute. The attributes could be divided in two groups. The product usage attributes defined the product portfolio of an individual customer, so these attributes can be considered to be internal (company owned), behavioral attributes. We also purchased external socio-demographic survey data that had been collected on zip

code level. All customers belonging to the same zip code area have the same value for these attributes. This included information on education, religion, marital status, profession, social class, house ownership and income. The selection of attributes to be used was made based on expert domain knowledge and exploratory data analysis (correlation with attributes to be predicted).

A number of preprocessing steps were taken, some of which are provided directly by the data mining environment we used for all the experiments (DataDetective, see www.sentient.nl). Most numerical attributes were transformed to categorical values. For each attribute, normalization factors were computed so that all attributes had the same standard deviation. Missing values were identified so that the algorithms that were going to be used could handle these values correctly.

Response Modeling

To select prospects we constructed a model to predict the likelihood of owning a caravan policy given all other attributes. Note that because of practical limitations, this was a simplification of the ideal model, which would have measured the response to a test campaign for a random selection of customers, or an alternative approximation in which the outcome would be propensity to buy a policy in the next n months. The overall response rates may be higher than the real response on a direct marketing campaign, given that ownership has been built up over time, and one must be cautious to interpret correlation directly as causation ('leaking predictors').

A random sample, the training set, was drawn from the customer base. The training set was used to construct a so called naive Bayes model. We will only provide an informal description here, see Witten & Frank (2000) for a more formal textbook description. In a naive Bayes model, the prediction for a given customer is computed by using the Bayes rule for statistical inference. This rule states how the probability of a class given data (attribute values for a test instance) can be computed from the probability of data given a class, the prior probabilities of the classes and the data (as derived from training data).

For instance let us assume that one of the input attributes defines a customer segment a customer is in. Now given a test customer with segment equals young professional we can derive the probability of owning a caravan policy by calculating on the training data, amongst others, the probability of being a young professional given that the customer owns a policy. The resulting estimates from each attribute are combined into a single score by assuming independence across attributes. This assumption is typically violated in practice, however, as long as the resulting predictions are interpreted as rank scores rather than absolute probabilities, naive Bayes generally delivers robust results.

A number of attributes were assigned very low importance, so the actual number of attributes taken into account to compute the resemblance was reduced to ten attributes using a subset attribute selection method (Correlation Based Feature Subset

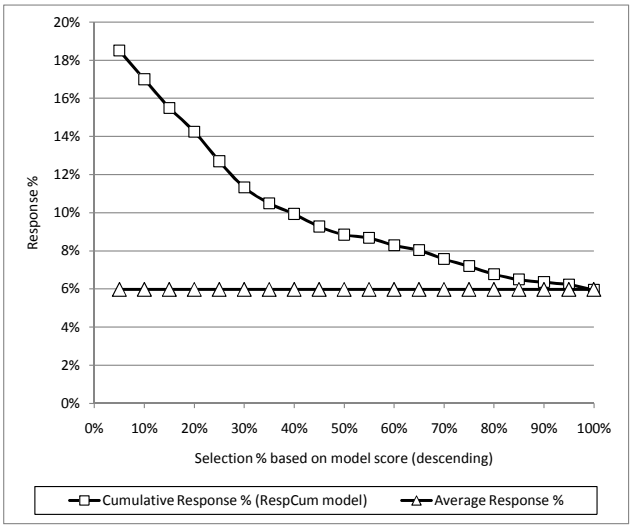


Figure 2.2: Cumulative response chart insurance model.

method (CFS) with best first forward search (Hall & Holmes 2003), (Hall 1999)). The prediction model was applied to the test set, a random sample of customers, disjoint with the training set. For each test customer, the response score was computed to indicate the potential to buy a caravan insurance.

Given a low response rate of 6.0%, a naive prediction model which scores all records as non-respondents, already achieves 94.0% classification performance. So this standard data mining measure, which counts the relative numbers of cases for which the classes were predicted correctly, did not suffice. This suggests that other evaluation criteria were needed to evaluate the accuracy of the prediction model.

For this kind of analysis, often a cumulative response chart is used (see figure 2.2). All test instances (records) are ordered from left to right on the x-axis with respect to their predicted probability of response (or a concordant rank score). If only the top 10% is mailed, the cumulative response rate $RespCum_m$ (relative number of respondents in mail selection) is 17%, which is almost 3 times higher than the response rate $RespCum_r$ achieved when records are selected randomly. At 25% the cumulative response rate is still more than twice as high than average (12.7%).

Another way to evaluate the model is shown in figure 2.3. Here the relative part of all respondents that is found is plotted:

$$RespCaptured_m = \frac{RespCum_m * s * n}{RespCum_r * n} = \frac{RespCum_m * s}{RespCum_r} \tag{2.1}$$

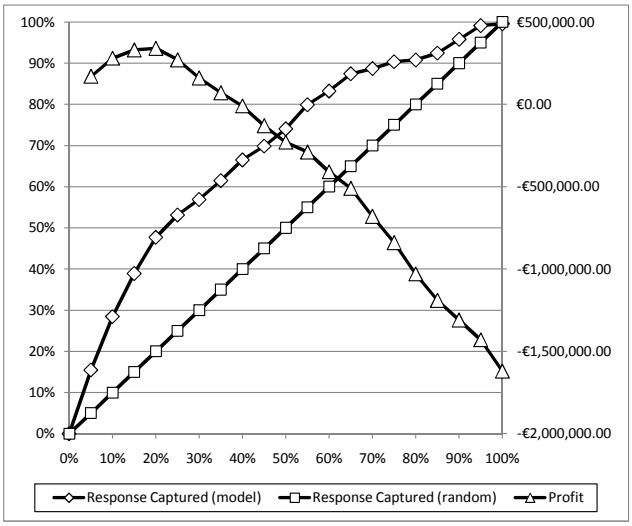


Figure 2.3: Captured response (left axis) and profit (right axis) insurance model.

with s selection size and n total number of customers in the test set or deployment set to select from. If the top 20% customers are selected, almost half of all respondents (48%) are found. The optimal mail selection size s^* depends on the cost per mail piece c and the profit per responder p. Profit (or loss) at s^* can be computed as the profit made on responders minus the costs of contacting the selection:

$$Profit_{s^*} = p * RespCum_m * s^* * n - c * s^* * n \tag{2.2}$$

with p the profit per responder (excl. campaign costs) and c the cost per contact. See figure 2.3 for an example with p=10 Euro, c=1 Euro, n=4,000,000 customers. Note this is for illustration purposes only, given the remarks made at the start of this section with respect to the outcome definition.

Descriptive Segmentation of Respondents

Contrary to prediction, descriptive data mining results cannot always be translated into measurable business results and interpreted in a single, objective manner. There are few generally accepted algorithm-independent error measures for segmentation techniques, and even less for profile discovery. In addition the business value resulting from descriptive data mining relies more on the way how the marketeer interprets the descriptions, the conclusions that are drawn and the actions that are taken, which is typically subjective and may be hard to measure.

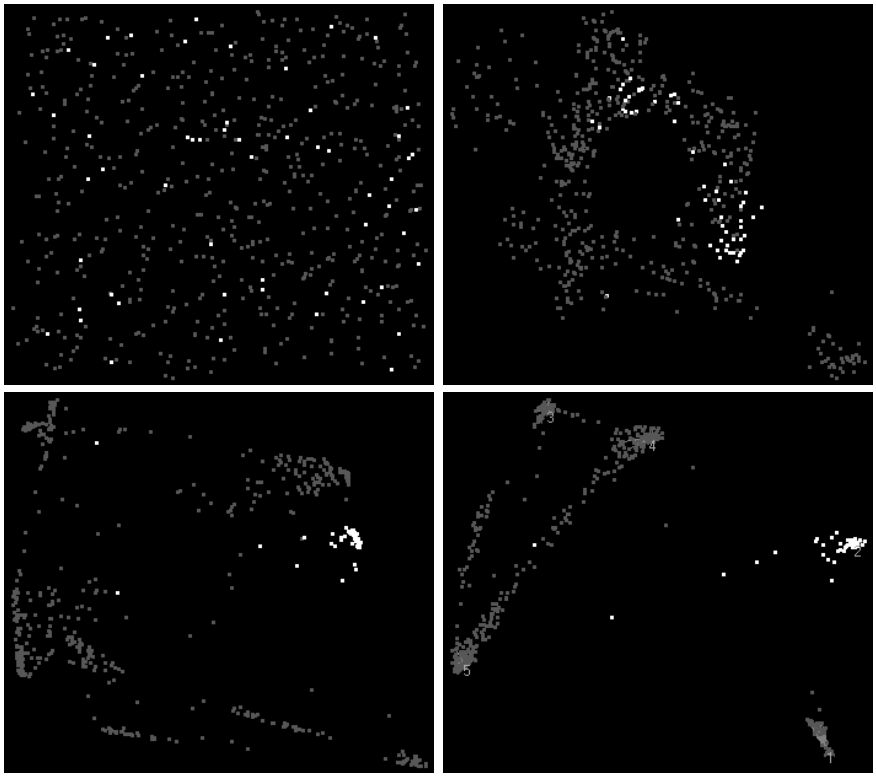


Figure 2.4: Iterative nearest neighbor projection reveals clusters in data. The figure at the right hand bottom side shows the end result. The user assigns the points to clusters depending on the final state of the projection and the resulting customer profiles. The projection process can then be rerun from a new random initialization point to cross check the stability of the projection and manual clustering. The white points belong to cluster 2.

In the insurance example, we wanted to find segments of caravan insurance owners which should be marketed with a different tone of voice, across different channels or using different offers etc. For this we used a custom iterative projection algorithm based on the principles of metric multidimensional scaling (Borg & Groenen 2005) as implemented in the data mining tool used. The algorithm projects all owners from high (95) dimensional space on to two dimensions in a number of steps (see figure 2.4). The process is started by randomly positioning customers in two dimensional space. Then in each step, every customer performs a nearest neighbor search in high dimensional space to find similar customers and moves a little bit in the direction of these neighbors in two dimensional space. A centrifugal force ensures that all customers do not end up into a single spot. In the insurance example, after the projection process converged, we built customer profiles of the resulting clusters. For instance, if we compared cluster 2 to all customers and the other clusters, we found out that the cluster contained relatively loyal customers (two car insurance policies, more often in a high turnover category), who where living in residential areas with a high rate of privately owned houses and who were more often belonging to a religious, senior citizen family structure.

2.1.6 DMSA Direct Marketing Cases

In this section we will propose some best practices for data selection and data mining algorithm selection, based on data mining experiences from a variety of direct marketing projects.

Developing a single data mining algorithm that offers the best peak performance (e.g. accuracy) on all possible data mining problems might seem a good approach for research. However the 'No Free Lunch Theorem', suggests that such an approach will probably not be successful. This theorem states that measured over all possible data sets and cost functions, there can be no single algorithm that performs best (Wolpert & MacReady 1995). So it makes more sense to identify the requirements for which a certain class of algorithms performs better. Furthermore, researchers sometimes assume that the data set is a given. However, the choice of data is probably even more important than the algorithm used. In the specific context of marketing, it might be possible to develop best practices for collecting appropriate data.

These issues have been the focus of a research project Sentient Machine Research has performed in co-operation with the Dutch Organization for Direct Marketing, Sales Promotion and Distance Selling (DMSA) (Wagenaar 1997). To be more exact, the objective of the research was to identify under what circumstances and for which data the relatively 'new' data mining algorithms such as neural networks, rule induction and evolutionary algorithms performed 'better' compared to classical statistical techniques such as linear regression and discriminant analysis. Experiments were performed in eight real world data mining projects for several organizations, includ-

Organization	Sector	Business Objective
Fund 1	charity	segmentation fund members
Fund 2	charity	upgrading, estimation donation potential
NS	transport	identifying suspects for prolongation of railways discount card
ABN AMRO	banking	selecting prospects for a new service (balance check by telephone)
VSB	banking	cross selling saving programs to current account holders
Readers Digest	publishing	selecting prospects from a list broker database
Centraal Beheer	insurance	selecting potential converters from a list broker database
NV Databank	list broker	selecting prospects for marketing of a new product

Table 2.1: Cases in the DMSA project.

ing banks, insurance companies, publishers, a railway company and charities (see table 2.1) (Wagenaar 1997). Below, we would like to share some of the lessons learned from this project.

Collecting the right data

A general result was that the data used was often the most important factor for the success of the data mining projects, in terms of both benefits and effort. Data collection was always a critical constraint in project planning. Data preparation amounted to up to 80% of total work invested. The requirement for what kind of data to use depends primarily on the nature of the data mining task at hand.

For prediction tasks, the data should possess as much predictive power as possible. Firstly, the number of attributes plays an important role. The more attributes are used, the higher the probability becomes that strong predictors are identified, and non-linearities and multivariate relationship can occur that intelligent techniques can exploit. On the other hand, the so called 'curse of dimensionality' limits the amount of attributes that can be used. If the number of attributes increases, the density of the data set in pattern space drops exponentially and complexity of models can grow linearly or worse (Bishop 1995). Complex models (i.e. a large number of parameters) have a higher chance of overfitting to the training data and will not perform well on new data (low generalization), so attribute selection is important.

Secondly, the type of attributes to be used is of importance. The best data to use is company internal behavioral customer data which relates directly to the products to be marketed or the customer behavior to be predicted. Examples are product usage, information requests, response, account balances etc. Traditional marketing

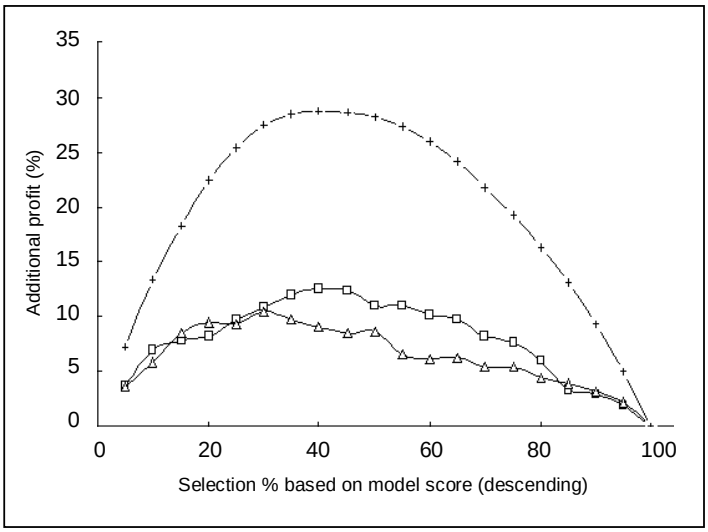


Figure 2.5: Chart with gains of several projects. Using company internal data on past customer behavior results in higher gains for response prediction tasks.

attributes such as social class, education, age and gender do not suffice to predict modern customer behavior. Empirical support for these claims is shown in figure 2.5. Highest gains were achieved in projects based on data as described above. For description tasks, however, raw summaries of product usage etc. do not suffice to inspire marketeers. Descriptive attributes, such as socio-demographic attributes that are collected at zip code level, should be added. These attributes typically possess much less predictive power, but offer more insight to marketeers.

Choosing the right algorithm

In the DMSA project, we roughly distinguished between adaptive pattern recognition techniques such as neural networks, rule induction, nearest neighbor and genetic algorithms and classical linear statistical techniques such as regression and discriminant analysis. The main advantage of adaptive techniques in general is that these techniques are able to model highly non linear relationships. Furthermore, it is often claimed that these algorithms are less parametric, i.e. make less assumptions about the relation between customer attributes and predicted behavior. For linear regression for instance, this relation is assumed to be linear, which is a pretty tough assumption. However, adaptive techniques require implicit assumptions about the data and relationships to be modeled as well. Also, real world marketing data is often

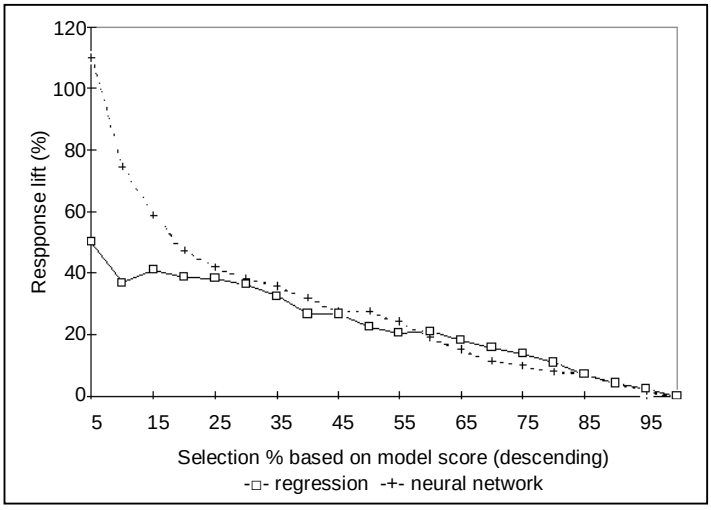


Figure 2.6: Creaming the crop with backpropagation neural networks. This graph displays the lift $(RespCum_m - RespCum_r)/RespCum_r$ for linear regression versus backpropagation neural networks for one of the DMSA projects.

quite noisy, in which case it is dangerous to try to fit complex non-linear relations, as one may run the risk of actually fitting noise rather than signal. Practical aspects such as speed, amount of data preparation needed, understandability and ease of use and deployment are also important. For example, linear regression is still one of the fastest algorithms around when it comes to scoring large data sets, and the resulting models are easier to explain to marketing stakeholders.

Overall we found that when the number of attributes was large enough for non-linearities to occur, and selection size was sufficiently small, neural networks in selected cases performed best in 'creaming the crop': selecting the top customers in a prediction task (figure 2.6). We achieved cumulative response rates in top selections that were up to twice as high as the runner up algorithms, which can correspond to considerable savings. For larger selections, the advantage of neural networks diminishes. This is reasonable to expect, because the relation between customer attributes and response for customers that have an average score is very weak. Also improved performance is certainly not guaranteed as overall performance not only depends on the ability to discover complex relationships, but also on robustness of the learner related to the levels of noise in the data. So if the marketeer is mainly interested in two-way, high quality relations with top customers, which are likely to be interested in the offer, including adaptive techniques for data mining may make

sense. If a marketeer is more interested in performing larger mailings to achieve a high response in absolute numbers, wants to mail a relatively large proportion of the customer base, and is willing to accept many non responders, classical techniques such as linear regression will suffice and may be easier to apply.

2.1.7 From Data Mining to Knowledge Discovery

The goal of data mining should be the transition to a learning, customer centric organization. Customer analysis should be performed on a regular basis, or customers should even be monitored on line. Prediction should not be limited to response analysis, but should be a business activity aimed at modeling higher level customer attributes, needs and attitudes, such as 'willing to take risk', 'high probability of ending the relationship' or 'early adopter'.

Reality is still different. A general problem of most data mining projects is that data collection is a major bottleneck, data mining algorithms still require a lot of manual parameterization and data preparation and reusability of data mining results is poor. To shorten the knowledge discovery cycle we suggest two important directions for research: automating data mining techniques and integration of data mining in an open knowledge management framework.

Automating Data Mining Algorithms

Although current data mining algorithms generally require less parameters to be set and less data preparation to be performed than classical statistical algorithms such as linear regression, users still need to have a low level understanding of how a specific algorithm works. We identify several possible directions to solve this problem.

First, in a practical approach, one could identify heuristics for making reasonable choices for specific applications, data mining techniques and steps in the knowledge discovery cycle. These best practices could be properly documented in some kind of data mining methodological framework, or ideally, these best practices are incorporated in intelligent assistants which guide the user through data preparation and prediction processes.

There are also less heuristic and more general algorithmic approaches, which are sometimes referred to as meta learning methods. These methods learn to make choices which were normally made by the data mining analyst, including deciding on the best algorithms to use (Aha 1992), (Soares & Brazdil 2000), (Vilalta & Drissi 2002).

Combining Knowledge Management and Data Mining

An important lesson from cognitive psychology is the so called Learning Paradox: 'He who knows nothing can learn nothing'. Whereas it might be fruitful to aim at automating the data mining algorithms, research into the direction of a more

open and integrated framework for storing and reusing data mining processes and results would be relevant as well, to further facilitate knowledge discovery driven institutional learning. There have only been a few attempts to introduce knowledge management to data mining (Wirth, Shearer, Grimmer, Reinartz, Schlösser, Breitner, Engels & Lindner 1997), (Engels, Lindner & Studer 1997). In our view research must not focus on a single analysis or algorithm, but on a number of analyses in a learning organization. For this it is required that analyst and domain knowledge is represented, a library of data mining results is maintained and intelligent assistance is offered based on this knowledge base.

2.1.8 Conclusion

Data mining can be a helpful tool for managers and organizations to cope with a dynamically changing and complex business environment. We identified best practices for application of data mining for direct marketing, selection of data and algorithms and evaluation of results. The key to successful application of data mining will be integration into business processes and information infrastructure .

2.2 Head and Neck Cancer Survival Analysis

The Head and Neck Cancer case is a second example of introducing data mining to an audience with no data mining or computer science background. The goal in this case is to predict five year survival probability for head and neck cancer patients. So called evidence based medicine is becoming more and more important in the medical field, from empirically based studies towards medical decision support systems.

We benchmark a wide variety of classification algorithms on this problem, resulting in varying accuracies. Whilst this may be sufficient to solve the problem at hand, this doesn't provide more insight from a data mining point of view *why* some classifiers perform better than others. Therefore we carry out a so called bias variance analysis to get a better idea of the source of the error (van der Putten & Kok 2005).

2.2.1 Introduction

Today an increasing variety of patient data is becoming available and accessible, ranging from basic patient characteristics, disease history and standard lab tests to micro-array measurements. This offers opportunities for an evidence-based medicine approach to diagnosing and treating head and neck cancer patients.

All this raw data does not necessarily equate to having useful information, on the contrary, it could lead to an information overflow rather than insight. What doctors need is high-quality support for making decisions. Data mining techniques can be used to extract useful knowledge from clinical data, to provide evidence for and

thus support medical decision making. In this section we will give a non-technical overview of what data mining is and how it can be applied in the head and neck cancer domain.

Let us consider survival rate prediction for head and neck cancer patients. When building a prognostic model no explicit medical hypothesis is made about the relation between the data items collected and survival rate. The task of finding the relation is left to a modeling algorithm. The medical analyst building the model then uses medical expertise to determine whether the patterns found are truly relevant to the prediction or perhaps a consequence of the particular way the data as been collected, data pollution or just a random effect.

Even if regular statistical techniques such as logistic regression are used to build the model, this example can be seen as a data mining project. For instance, the focus is on knowledge discovery rather than confirming hypotheses. Furthermore the patterns found must be useful for medical decision support.

Within the cancer domain data mining is being applied for a long time already. Examples are the classification of breast tumor cells as benign or malignant, distinguishing different types of leukemia by mining micro-array data (Golub, Slonim, Tamayo, Huard, Gaasenbeek, Mesirov, Coller, Loh, Downing, Caligiuri, Bloomfield & Lander 1999), (Liu & Kellam 2003) and predicting breast cancer recurrence (Michalski, Mozetic, Hong & Lavrac 1986). Recent developments in functional genomics and proteomics have been key drivers for the application and development of biomedical data mining. The major data mining conferences host specific workshops on biomedical data mining, for instance the BIOKDD workshops at the KDD conferences from 2001-2008 (see for example Lonardi, Chen & Zaki (2008)) and Bioinformatics workshops at ICML-PKDD (for example Ramon, Costa, Florencio & Kok (2008)). Data mining is not limited to simple data - the same process and techniques are used to mine imaging data and semi-structured data such as molecular structures, and a hot topic at the moment is the application of data mining to text such as medical articles.

Survival prediction is an example of a so called predictive data mining task. The goal here is to assign the right class to a patient, for instance dead or alive in five years. This is called a classification task, or the task is called a scoring task if the task is to produce a rank score reflecting the probability to be alive in five years. An alternative prediction task would be a regression task: the goal here is to predict some unknown continuous outcome, for instance the number of years that someone will live from now on. In both cases we need to have some data available on patients for whom the outcome is known.

2.2.2 The Attribute Space Metaphor

Let us explain classification in more detail using the concept of an attribute space (or also: pattern space). Assume the goal is to develop a five year survival model.

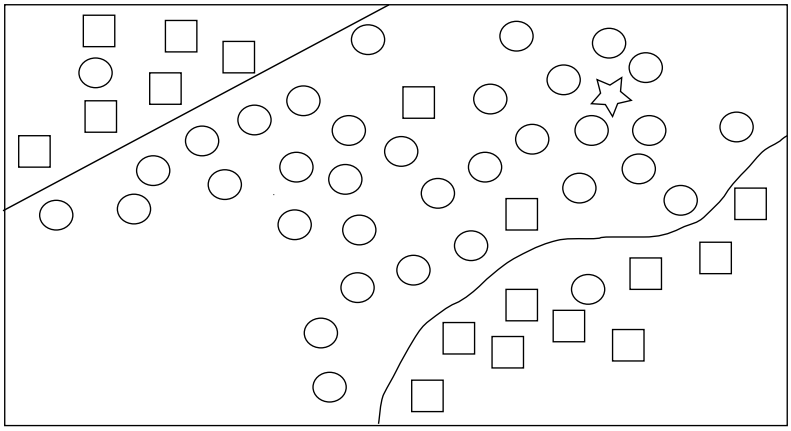


Figure 2.7: Classes 'square' (dead) and 'circle' (alive) in two dimensional attribute space. Each dimension corresponds to an attribute, for instance 'age' and 'tumor size'. The star corresponds to a patient for which a prediction needs to be made about probability of survival after five years.

To develop a model we have a data set of cancer patients available with a known outcome, deceased or alive five years after admission. For each patient a number of attributes (variables) are known that can be used to make the prediction, for instance age, location of the tumor, size of the tumor etc. So the patients can be seen as points with a certain location in attribute space, with each of the attributes corresponding to a dimension. Each of the points can be labeled with the outcome class: dead or alive in five years from now.

In figure 2.7 we have visualized this for two dimensions, assume that 'square' means the patient is dead and 'circle' alive after five years. It is now easy to see what the task for the classifier is: separate the classes in attribute space. In this example the classifier divides up the space in three areas, two of them correspond to the class deceased and one to class alive. For the new patient indicated by the star in the figure the classifier will predict class alive.

Note that the classes in the left upper corner are linearly separable: we can separate them with a single line. However, there are also some deceased patients in the lower right corner. So we can't separate the whole space with a single line. This means that for this example a classifier that creates a single (linear) decision boundary between classes is sub optimal. The simple binary logistic regression model is an example of a linear classifier; a wide variety of more advanced regression techniques are available that can model non-linear decision boundaries.

Also note the two deceased patients (squares) in the middle 'alive region'. In real data sets there will be a lot of overlap like this, and drawing different samples

from a data set will lead to different outcome class distributions in attribute space. The goal of the classifier however is not to model this particular data set, but rather the underlying mechanism that is generating this data: the relation between patient attributes and survival rate in this case. The classifier needs to strike the right balance between recognizing intricate decision boundaries and not overfitting the data. In some cases two patients with exactly the same attributes may conflict in terms of class labels. This is an example of a data point the (theoretical) optimal classifier cannot even handle, as it only has attribute information available to make its prediction.

There is a whole range of techniques available for building classifiers. We find the distinction 'statistical' versus 'data mining' not particularly useful (if even possible), we rather differentiate the techniques on the ability to model complex decision boundaries, how easy it is to interpret the model and the risk of overfitting data. For brevity we have excluded a discussion here, see van der Putten & Kok (2005) for a non technical comparison of various classification techniques such as nearest neighbor, neural networks and decision trees, using the attribute space as a common metaphor.

2.2.3 Evaluating Classifiers

Several procedures exist for evaluating the quality of a classifier; here we distinguish between internal and external validation methods. Generally it is not advised to use the entire set of known cases for training. Because of overfitting the risk exists that the classifier gives excellent results on the training data, but when it is applied to new cases the results are very poor. If all available data is used for training the generalization capabilities of the classifier cannot be tested. The simplest internal evaluation method is hold out validation. One part of the data is used to create the classifier, the other part is held out to test the performance of the model on cases that have not been used for training. A more sophisticated internal validation method is cross validation. In tenfold cross validation for instance, the data set is divided into ten parts. First a classifier is constructed using the first nine parts and validated on the tenth part. Then a classifier is built on the first part plus part three through ten and validated on the second part etc. This process is usually repeated over a number of runs. This procedure will result in a more accurate estimate of the model performance.

Generally several types of classifiers with varying parameter settings are tested out using cross validation, the best classifier is then chosen and retrained on the entire data set to yield a single model, and the patterns found by the model will be checked by a domain expert. External validation tests evaluate a classifier on completely different samples. For instance, a survival rate model discussed in Baatenburg de Jong, Hermans, Molenaar, Briaire & le Cessie (2001) was built on patients from the Leiden University Medical Center, but later applied to patients from other hospitals.

Special evaluation measures exist that decompose the error into different sources or causes, intrinsic error, bias and variance. For the purpose of brevity we will only describe these informally in this section (see Geman, Bienenstock & Doursat (1992), Kohavi & Wolpert (1996), Breiman (1996), (Friedman 1997), Domingos (2000), James (2003), van der Putten & van Someren (2004) for more formal accounts). The intrinsic error (or noise) is the unavoidable error, i.e. the error that some ideal classifier would still have, given the input data that is available. For instance, imagine that two patients have exactly the same attributes but only one of them dies. As discussed, even the ideal classifier would, using this data, predict the wrong class for one of the patients. The bias error is the error due to bias, i.e. limitations in the relationships that a certain classifier can express or find, even if an infinite number of instances would be available. For instance linear or logistic regression models with thresholding essentially create a single hyperplane in pattern space as a decision boundary (i.e. a line in 2d, a plane in 3d etc.) so more complex patterns will not be recognized. Finally the variance error is the error due to the fact that only limited data is available. Instability of a learner on a single data set or overfitting, different results for different samples from the same data set, will lead to increased variance error.

2.2.4 Leiden University Medical Center Case

In this section we will present some case results. Note that the scope and purpose of this section is to give an illustrating example of data mining rather than presenting a thorough medical, statistical or data mining analysis (Baatenburg de Jong et al. 2001), (van der Putten & Kok 2005).

Objectives and data used

The objective in this case is to provide a prediction of the probability of survival over the full range of the next ten years. This corresponds to the main question a patient will have – how much time do I have left, or what is the probability that I still will be alive in x years. Special statistical survival regression techniques exist to create models to answer these questions (Harrell 2001). However, to simplify the explanation of the classification algorithms and the benchmark experiments, we approximated the objective with the more basic task of classifying whether a patient will be deceased or alive after five years.

The data set we used was a variant of the data set from Baatenburg de Jong et al. (2001). It contains 1371 patients with head and neck squamous cell carcinoma of the oral cavity, the pharynx, and the larynx diagnosed in the Leiden University Medical Center (LUMC) between 1981 and 1998. From these patients, the prognostic value of site of the primary tumor, age at diagnosis, gender, cancer staging (T-, N-, and M-stage), prior malignancies and ACE-27 (co-morbidity, i.e. an indication of overall physical condition) were known. Patients were staged according to the

UICC manual and prior malignancies are defined as all preceding malignant tumors except for basal cell and squamous cell carcinoma of the skin. If contact with the patient is lost there is an independent and active follow-up by contacting the family doctor and reconciliation with the Dutch Registry of Births, Deaths and Marriages. This guarantees that the outcome (dead or alive at a given stage) is as complete as possible.

We experimented with two versions of the data set, depending on how we treated the TNM cancer staging data. TNM is a cancer staging system to assess the extent of cancer in a patients body. T measures the size of the tumor and whether it has invaded neighboring tissue, N describes regional lymph nodes affected, and M describes distant metastasis (spread of cancer between parts of the body). In the first data set T, N and M were measured as separate numerical attributes. In the second data set T, N and M were grouped into symbolic TNM categories, e.g. T2N0M0.

Modeling approach and results

To gain experience with this data set a wide variety of classifiers have been tested including logistic regression, nearest neighbor (with 1 and 15 neighbors respectively), decision trees, decision stumps (trees with only a single split) and neural networks (single hidden layer, decaying learning rate); see van der Putten & Kok (2005) for a description of these methods in the context of the head and neck cancer case. Furthermore we have added some other classifiers: support vector machines, naive Bayes, decision tables and a bagged decision trees ensemble. All classifiers have been tested on the two data sets (numerical versus symbolic TNM, see above) with ten runs of tenfold cross validation: in total 2000 classifiers have been built. We used the WEKA open source data mining package for the experiments (Witten & Frank 2000). To simulate a real world setting with time and modeling expertise constraints, and to avoid that the familiarity of the experimenter with certain algorithms would become a factor in the performance of the algorithms, we have used default settings unless stated otherwise.

In figure 2.8 an example of a decision tree generated from this data set is shown (C4.5 decision tree (Quinlan 1986) on the full set with confidence setting of 0.05). Note that T, N, age, ACE and prior malignancies have a role to play in this model, but M status surprisingly enough does not. We can only speculate, but apparently the first few splits divide the patient population into subgroups within which the M status does not appear any more as the top indicator, potentially because of strong correlation with other predictors appearing in the tree. For each of the leaves we have also calculated the proportion of deceased or alive patients, dependent on the class label of the leaf.

Tables 2.2 and 2.3 provide an overview of the average and standard deviation on the classification accuracies for each of the classifiers over all runs (TNM numeric versus symbolic data sets). Classification accuracy is defined as the percentage

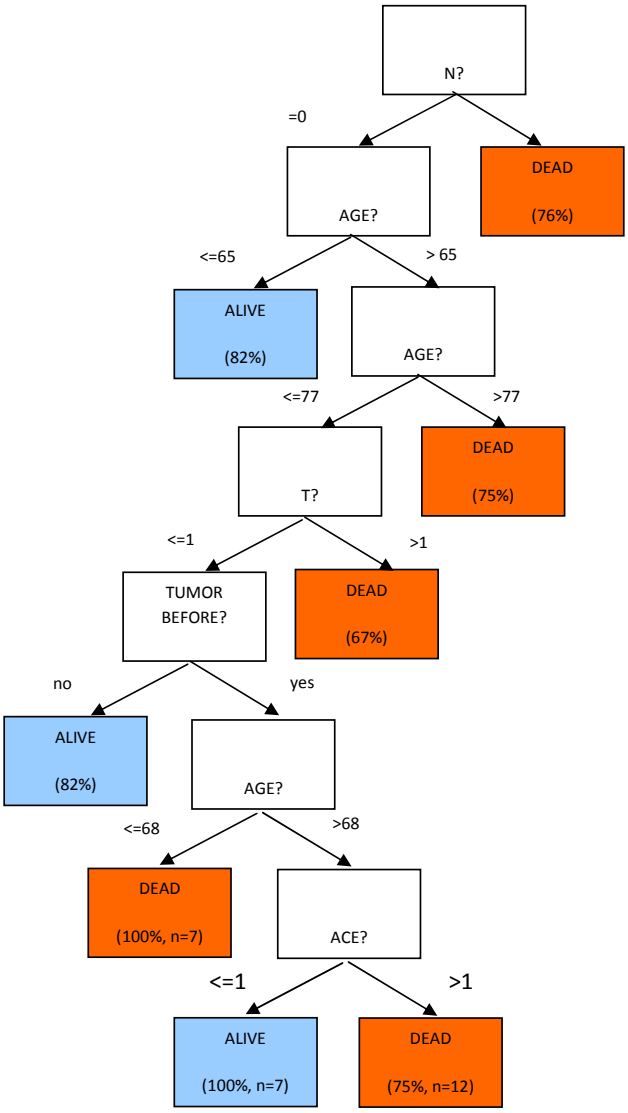


Figure 2.8: Decision tree generated from Head and Neck data (full set)

Classifier	Performance
Neural Network	73.0 ± 4.1
Logistic Regression	72.8 ± 4.0
SVM	72.3 ± 4.1
Naive Bayes	70.9 ± 4.2
Bagged D. Trees	70.0 ± 3.8
Decision Tree	69.4 ± 3.8
Decision Table	69.2 ± 3.6
Decision Stump	67.6 ± 3.4
15 Nearest Neighbor	67.6 ± 4.1
1 Nearest Neighbor	61.1 ± 3.5

Table 2.2: Average and standard deviation on the classification accuracy for all classifiers (TNM numeric data set)

Classifier	Performance
Logistic Regression	71.1 ± 3.6
Neural Network	71.0 ± 3.7
Decision Tree	70.3 ± 3.8
Naive Bayes	70.2 ± 3.9
Bagged D. Trees	69.7 ± 3.9
Decision Table	69.6 ± 3.7
SVM	68.9 ± 3.8
Decision Stump	68.0 ± 3.7
15 Nearest Neighbor	66.5 ± 3.6
1 Nearest Neighbor	62.0 ± 4.4

Table 2.3: Average and standard deviation on the classification accuracy for all classifiers (TNM symbolic data set)

correct classifications on the hold out validation set. Note that the differences for most classifiers are quite small given the standard deviation.

The grouping of algorithms in terms of performance is interesting. For instance for the numeric data set (table 2.2) the top four classifiers are different but all compute some weighted function over the attributes to generate a probability score for the predicted class. The model formula is valid over the entire attribute space. The decision trees, tables and stumps and the nearest neighbor algorithms rather divide the attribute space up into small regions – the class prediction is generated from this local region only. It seems that on this data set the latter strategy is performing worse. In theory, naive Bayes can be seen as a borderline case of both approaches, given that its key model parameters are estimated for specific values of each attribute, and then an overall function is applied to these parameters. It is interesting to see that this is also reflected in the results as naive Bayes is the bottom ranked classifier in the top four. The models built on the numeric data set outperform the models built on the symbolic data; apparently these models can exploit the ordinal relationship of these attributes, this likely leads to more robust models.

To compare the classifiers in more detail we have performed a so called bias variance analysis. As explained in section 2.2.1, bias variance analysis assumes there are three potential sources of error, intrinsic error, bias and variance, informally the unavoidable error, the error due to model representation or search limitations and the error due to instability of the model over random samples.

In the particular bias variance computation procedure we used, the intrinsic error is combined with the bias error ((Kohavi & Wolpert 1996); TNM numeric data set, bias variance decomposition sample size 350). Figures 2.9(a) and 2.9(a) show the results sorted by bias and variance respectively. Classifiers with high variance are Decision Stumps, -Tables, -Trees and 1-NN, high bias classifiers are Naive Bayes, Neural Networks, 1-NN and Logistic Regression. Logistic regression, Naive Bayes,

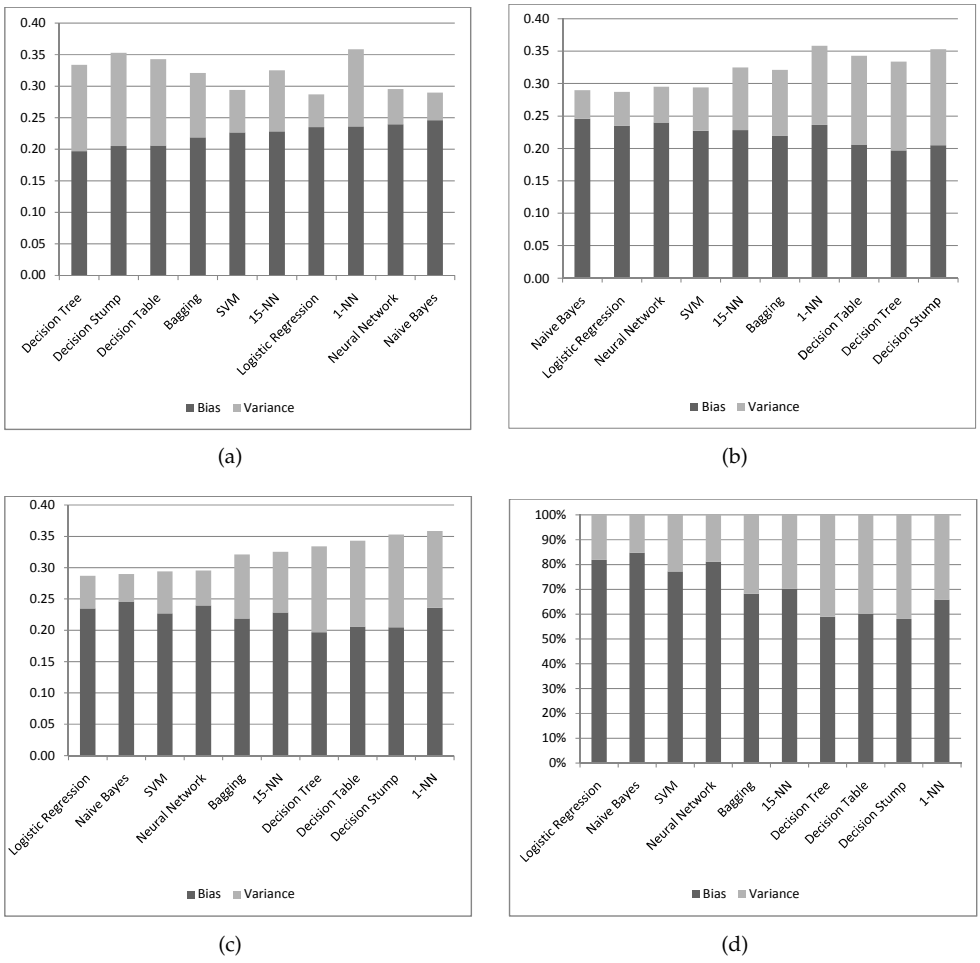


Figure 2.9: Bias variance analysis on HNC data: (a) sorted by increasing bias; (b) sorted by increasing variance; (c) sorted by increasing total error (bias plus variance); and, (d) same sort order but stacking to 100% to show relative proportion of bias versus variance.

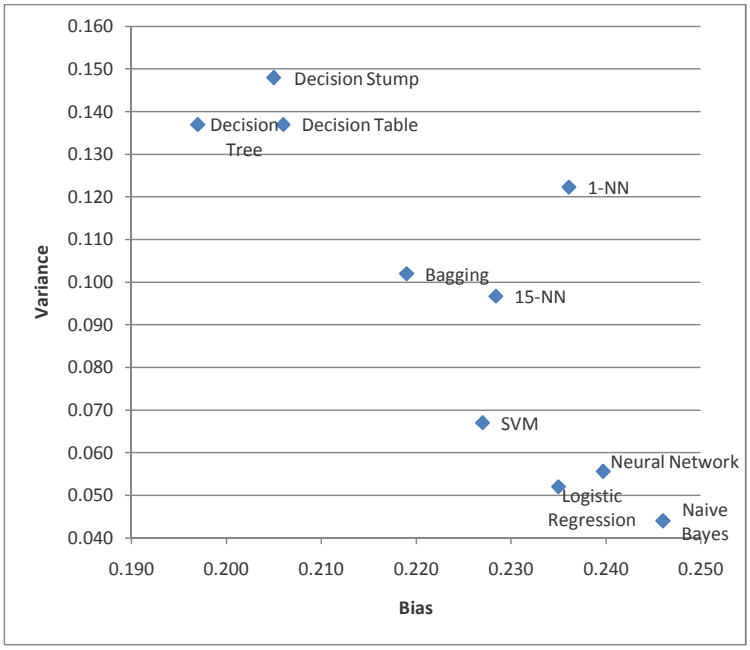


Figure 2.10: Plot of bias versus variance for a variety of classifiers on the HNC data set (TNM numeric). Note the the difference in scaling between the bias and variance axis.

SVM and Neural Networks perform well on this data set (figure 2.9(c)). According to figure 2.9(d) the bias component error is largest, however it should be taken into account that this includes the intrinsic error, which is unknown but constant across the classifiers. As can be seen the variance component in relative terms generally increases when total error becomes larger.

To better visualize the relationship between bias and variance we have also created a bias variance scatterplot (figure 2.10). The tradeoff between bias and variance can be clearly seen here. The tree based classifiers have a lower bias error than for instance logistic regression, but this is completely countered by the variance error, which is up to three times larger for the lower scoring classifiers than for the top classifiers. In more detail, the standard deviation for the bias component is 0.017, whereas the standard deviation of the variance component is 0.039, so variance is a more important source for the difference in error across the various classifiers. We have also performed a linear regression against these points, this results in the following equation:

$$variance = -1.98 \times bias + 0.54 \tag{2.3}$$

Obviously this is a somewhat artificial analysis, given that this equation will not be valid across the entire bias variance space. Ignoring this for a moment we can perform some speculative yet interesting calculations with this equation, for instance providing an estimate for intrinsic error. Remember that in the method used, bias includes intrinsic error and the real bias (Kohavi & Wolpert 1996), so it will never become 0. The (theoretical) ideal classifier will have intrinsic error, but no real bias and variance (the Bayes error, see van der Putten (1996) for a description). Solving the equation for zero variance results in a bias estimate of 0.54/1.98 = 0.27. Given that the real bias will be zero, this provides us an estimate of intrinsic error. i.e. the error for the ideal classifier, under the assumption that the tradeoff relationship would be as per the regression equation.

2.2.5 Discussion and Conclusion

In this section we have presented an introductory overview on the application of data mining for head and neck cancer survival analysis. We simplified the survival analysis problem to a classification problem and benchmarked a range of classification algorithms on the resulting data. To get a better idea why certain classifiers are performing better than others we evaluated the various sources of error, bias and variance. Variance is the main cause of differences across classification methods.

Biomedical data mining is one of the fastest growing areas in knowledge discovery. While the field will be maturing we expect a shift of interest from improving the core classification algorithms to improving the wider applicability of data mining. This may include more emphasis on automating and supporting the full data mining process rather than just the core modeling step, generalizing data mining methods to more types of information from structured to more unstructured data and building data mining and decision support systems that can easily and reliably be used by doctors or medical analysts not data mining experts. The latter step will require more efforts need to be made to blend predictive models for instance with existing medical knowledge, policy rules and protocols.

The classification algorithms described output class labels (deceased or alive), but can also generate probabilities, for example the probability to be alive after five years. As discussed, special purpose regression techniques exist for survival analysis, for example Cox regression. One of the advantages of these techniques is that they can take the information of censored patients into account: patients for which the outcome (dead or alive at year x) cannot be completely determined (Harrell 2001). A lot of work can still be done to adapt the existing data mining techniques further specifically to the problem of survival analysis, as has been done in the statistical community.

A specific research question that we are interested in is how to integrate information from different sources. For instance, for prediction of 5-year survival rate

various data sets are available from different institutions. Is it possible using this data to develop a prognostic procedure that produces more reliable results than can be obtained by just using one of these single sets alone? Would this be a matter of combining information on a data set level by just adding the data sets together or using a more sophisticated data fusion procedure that fills in information that is missing in some of the data sets (van der Putten, Kok & Gupta 2002*b*)? Or would a better approach be to build separate prognostic models on each single data set and combine the outputs of these models? Some of our initial experiments seem to indicate just adding data sets together is sufficient if there is a lot of variety between data sets, but this may be different if there is more standardization across these sets (Maat 2006). Given that variance seems to be a major component of error the use of more data may result in improved classification models, in terms of accuracy and robustness.

2.3 Detecting Pathogen Yeast Cells in Sample Images

This case reports on an approach towards developing classifiers for detecting virulent cells in a yeast sample, by using a range of features derived from the shape or density distribution in an image. The classifier can be used for automating screening and annotating existing image collections. A purpose of this case is to show an example for which the core modeling step clearly is just a small part of the overall process and problem, and choices made outside the modeling step can have a major impact on the overall success.

We will describe the full end to end process from growing yeast samples, capturing images, derivations of features, supervised and unsupervised data mining and evaluation of the results. We compare various expertise based and fully automated methods of feature selection, benchmark a range of classification algorithms and in general, illustrate the application of data mining to this particular domain.

For one of the problems in this case we demonstrate that all classifiers perform roughly the same - almost perfect performance. In practice, it can occur quite frequently that problems are either trivial or too hard to solve, whereas in typical machine learning papers problems are carefully selected to be not too hard nor too easy. That said, it is still an open question whether the underlying problem is hard to solve (classifying yeasts), whereas the data mining problem is easy (classifying pictures). In our opinion, this is a good example that in practice, the translation of the research or business problem to a data mining problem has a major impact on the results (Liu et al. 2006), (van der Putten et al. 2007).

2.3.1 Introduction

Yeast cells come in many appearances, yet only few of them have been identified as being pathogenic. The virulence of a yeast cell can, in some cases, be derived from the morphology of the cell. In the pathogenic yeast Cryptococcus neoformans thicker capsules are believed to be an indicator for virulence for instance. The aim of our study is to develop a measurement and classification system for the virulence of cryptococcal yeast cells, using their morphological characteristics, not limited to capsule thickness alone. The classification is initially based on image features, but it should be possible to extend the procedure in multi-media-like fashion to include biochemical and genomic data. By definition, an image classification system depends on the quality of images and derived features that are fed into it. We therefore take a rather holistic view on the construction of such a classification system.

Cryptococcus neoformans is a basidiomycetous yeast that can cause meningitis, meningoencephalitis and pulmonary and skin infections. Infections occur mainly in immunocompromised patients, for instance HIV-infected patients, transplantation patients and leukemia patients (Casadevall & Perfect 1998). One of the most significant virulence factors of the fungus is the presence of an extra-cellular polysaccharide capsule (Littman & Tsubura 1959), (Dykstra, Friedman & Murphy 1977), (Chang & Kwon-Chung 1994), (Bose, Reese, Ory, Janbon & Doering 2003), (Janbon 2004). Complementation of a capsule-deficient mutant clearly showed the relation between the presence of a capsule and cryptococcal virulence (Chang & Kwon-Chung 1994). The thickness of the capsule can vary between strains, specific genetic constructs related to capsule biosynthesis, and between different environmental conditions (Dykstra et al. 1977), (Casadevall & Perfect 1998), (Zaragoza, Fries & Casadevall 2003).

Although measuring the size and shape of the capsule seems straightforward, it has not often been applied. Using the morphology of the yeast cells, the obvious analysis is to look at the capsule thickness directly, either by automated or semi-automated methods (Rivera, Feldmesser, Cammer & Casadevall 1998). Early attempts may have been hampered by the fact that the staining methods were not sufficiently developed for good image analysis in that the staining results were not reproducible. We have experienced such as well in earlier work (Liu et al. 2006). Newer staining methods have opened possibilities for large scale analysis.

Applying an image analysis driven method will allow for deriving more features than just the capsule thickness. Rivera et al. (1998) have analyzed samples of mouse brain and lung infected with C. neoformans to evaluate both capsule thickness and cell volume. The cells were segmented from the images using hand-tracing. The features were derived by estimating the radius and computing the features with the analytical equations of circle area and sphere volume.

However, a fully automated tool has not been presented to date. Given the relatively simple shapes of cryptococcal cells, an effort to develop such a system should be undertaken. The extracted features from cell images should inform us

about the different classes of cryptococcal cells that can be distinguished. Ideally, a sample is taken from a population, i.e. a yeast culture, and from the features a distribution of the virulence can be found leading to a further understanding of the virulent state of a particular culture. Features should not be restricted to capsule thickness, but extend to a broader range of features that can be extracted from images of cryptococcal cells.

Such a fully automated approach of image analysis and classification is relatively new in the field of yeast genomics. It is a desired approach though, as in the near future such systems will be used in large scale screens. This case presents a first step towards such a system, with a focus on an analysis of the relative importance of the derived features. Features and images should be stored in a database, and interoperable screening will allow retrieving other features related to the same sample from other databases. This is the typical trend currently seen in the bioinformatics research, i.e. an integrative approach of bioinformatics combining information from a broad panel of related bio-medical, molecular and organismal databases. In summary, this case presents a holistic description of an end to end process from growing the yeast, capturing images, image segmentation, feature extraction and classification.

2.3.2 Materials and Methods

The pipeline for experimental data consists of a range of modules passing files and data (see figure 2.11). The process starts with producing the yeast strains followed by image acquisition. The images need to be further enhanced first, and then separate cells are isolated through an image segmentation procedure. For each of the cells, image features are derived and then fed into various data mining algorithms for classification and clustering. Below we will describe each of these steps in more detail.

Producing the Yeast Strains

Preliminary investigations were made using a variety of Cryptococcus strains from the collection of the CBS Fungal Biodiversity Centre, using some media that are known to influence capsule size. Among these were Littman medium (Littman 1958), Golubev medium (Golubev & Manukyan 1979) and the recently described Sabouraud media with or without MOPS, HEPES, pH5.5 and 7.3, and 1/10 diluted Sabouraud medium (Zaragoza & Casadevall 2004).

Unfortunately, results obtained using these growing conditions were not optimal. Therefore, we decided to use Potato Dextrose Agar (PDA: 230 ml potato extract, 20 g dextrose, 15 g agar, 770 ml water, pH 6.6) that according to our experiences at CBS result in highly mucoid colonies and capsulated yeast cells in many basidiomycete yeast species. Two C. neoformans strains were used in the final analysis, namely an acapsular mutant CBS 7926 (Cap 59- mutant of NIH B-3501, E.S. Jacobson) and a

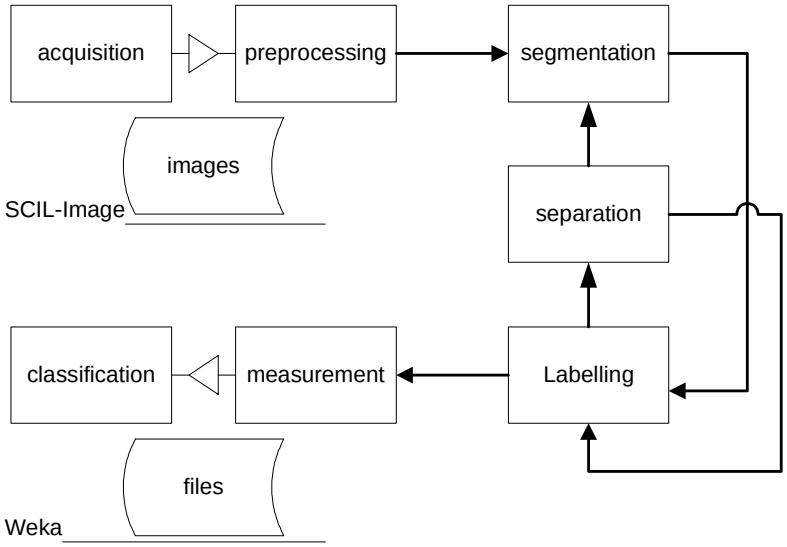


Figure 2.11: The system process flow. The thick lines indicate the flow of the data as they are passed to another module. The open arrows indicate the flow of files (image in and measurement file out) as used in the system. ScilImage is used for all image related work; Weka is used for work related to the classification and clustering

capsule containing isolate CBS 7936 (Cap 67- mutant of NIH B-3501, E.S. Jacobsen). The strains were maintained in the gas phase above liquid nitrogen at -135C, subcultured twice at PDA (48 h at 25C) and investigated for capsule size. In addition CBS 6955 (=ATCC 32608 = NIH 191), a Cryptococcus gattii strain, was used in our experiments. This strain was cultured on the standard YPGA medium (1% yeast-extract, 1% peptone, 2% glucose, 2% technical agar 3, all in w/v); the staining procedure for this strain was equal to the other two strains in this study.

In order to enhance the contrast for the imaging, we used the nigrosine staining method for our experiments. All yeast cells (CBS 7926, CBS 7936 and CBS 6955) were stained in 9 μl of 5% (w/v) nigrosine in water. The yeast cultures were inspected under the microscope to inspect if the culture was indeed according to expectations. Growth of the yeast culture on agar-plates was tested with a Leica Stereo microscope. Cultures that were proven to be of good quality were used in the experiments and slide preparations were obtained.

Staining and Image Acquisition

In the image acquisition phase the focus was to obtain images of sufficient quality for images analysis. The starting point for this study was to use two dimensional images. From earlier research with cryptococcal cells we have learned about the quality of various staining methods. The staining methods that we have focused on are the so called background staining techniques that do not stain the specimen but rather enhance the microscopy features by making the background more light dense (Liu et al. 2006).

Traditionally, yeast biologists were using Indian Ink in the microscopy preparation. The apparent disadvantage of this staining method in image analysis and specimen classification is that it is hard to get a reproducible staining. In addition application of this staining results in blurring of capsule margins. With the nigrosine staining method we were able to get reproducible staining with few artifacts. The staining method also rendered an excellent quality of the visualization of the yeast cell and its capsule. Moreover, it can be applied as a standard procedure and therefore it is easily included in the standard workflow in a yeast biology laboratory.

Some examples of images that are obtained with the nigrosine staining can be found in figure 2.12. Each image is a sample and from each culture at least 20 images are obtained through random selection over the slide. The operator criterion is that in a selected field of view, sufficient cells are present. The samples were not taken to establish the numerical density of the yeast in the specimen preparation.

Figure 2.12 (a) includes cluttered cells that need be separated in the preprocessing phase so that each cell can be quantified. Figure 2.12 (b) illustrates a budding yeast cell. The 'new' cell is excluded from the analysis: in preprocessing the bud is separated from the parent cell and the parent cell is used in the analysis. Figure 2.12 (c) depicts a cell that is captured incompletely. Cells on the image border are

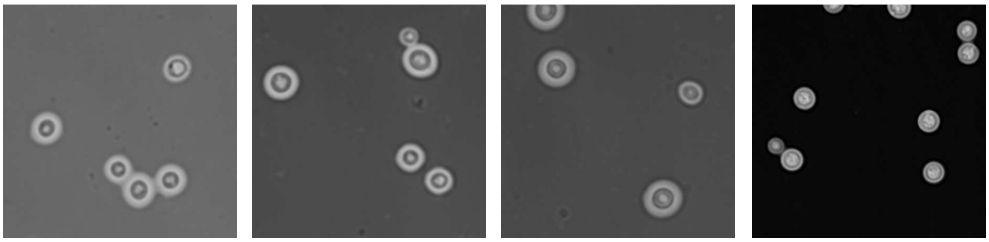


Figure 2.12: Four examples of samples taken from a yeast culture negatively stained with nigrosine.

excluded from analysis. Figure 2.12 (d) depicts another yeast culture with cluttered and budding cells as well as cells on the image border.

The slides were prepared for image acquisition on a Zeiss Axioskop with a PlanApoChromat 63x (NA 1.40) oil lens. The image acquisition was carried out with an Adimec MXI2P black and white CCD camera mounted on the Zeiss Axioskop (Zeiss, The Netherlands) and connected to a Pentium 3; the acquisition was controlled by the Research Assistant (v3) software. The Adimec MXI2P acquires images with a dynamic range of 8 bits; the images are sized 640x444 pixels and stored as TIFF files.

Image Preprocessing and Segmentation

Image preprocessing and segmentation is crucial to the experimental set up, given that it generates the results for the final classification procedures. The processing and analysis of the images is carried out using the SCIL-Image environment (van Balen, Koelma, ten Kate, Mosterd & Smeulders 1993). This image processing environment is an extensible software package well suited to scientific research. To complete the tasks of the research that is described a range of new routines were added to the package.

The preprocessing step consists of preparing the images for a segmentation procedure, so that each individual yeast cell in the image is measured through a number of features. Ideally the yeast cells are evenly distributed over the microscope slide and the yeast cells captured in images are nicely separated. This is, however, not always the case (figure 2.12 (a)-(d)). In the images we find budding cells, i.e., cells in the process of division, and cells that are cluttered together as well as dead cells. The classification of yeast cells should be based on features that are derived from single cells. The effort in the preprocessing and segmentation step is to accomplish that particular goal. This goal is achieved by applying firstly the appropriate filters (figure 2.13 (b) and (c)); secondly performing the segmentation and thirdly processing each

segmented image in such a way that only features from single cells are extracted. This requires application of heuristics in the segmentation process. An image is successfully processed if each of the cells that are completely visible in the image is transmitted to the image analysis part and its shape can be measured. Figure 2.13 (e) is supportive in understanding the segmentation procedure; a cryptococcal yeast cell consists of a cell body and a thick (or no) capsule. The segmentation is a multi-layered process as we have to be sure to extract the individual cells in the right way. Below we list the processes involved in the segmentation procedure in more detail.

The segmentation starts with a straightforward bi-level threshold operation, which results in a binary image with just the capsules. In this phase of the segmentation it is important that the outer contour of the cell (the outer boundary of the capsule) is detected accurately. Using propagation a mask is created over the area that contains the entire cell. Next, of all objects in the image, the objects that touch the boundary are established and by an XOR operation these are excluded from the binary image. In doing so, the objects on the boundary, often incomplete shapes, do not contribute to the measurements (for instance figure 2.12 (c) and (d)). The small sized objects are removed.

The next step is a labeling operation, so that we can extract and address each of the cells present in the image. In this phase we have to evaluate whether or not the cells are cluttered. For this evaluation the characteristic that intact individual cryptococcal cells are circular is used. This is accomplished with a circularity criterion which approximates one (1.0) for a circle. Cells that measure as circular are further segmented for measurement, cells with an aberrant circularity are processed in the separation module (see figure 2.11).

The circular mask is used to extract each of the labeled cells and perform a precise segmentation of the capsule (cf. *CC* in figure 2.13 (e)). This is completed in a buffer image of just size of the bounding box of the shape. The circular mask, obtained from the labeling is used in an XOR operation to find the area of the cell body (cf. *CB* in figure 2.13 (e)). After segmentation we have obtained two new masks, namely one for the capsule (*CC*) and one for the cell body (*CB*). In figure 2.13 (d) a segmentation result is depicted by superimposing the contours of *CC* on the original image. The masks and the buffer image are passed to the measurement module.

In case the circularity is not approximating 1.0, it is probable that the label represents a clutter of cells. An example of such a clutter is depicted in figure 2.14 (a) and in figure 2.14 (b) the binary labeling is shown. These cells need to be separated in order to be able to use them in the measurement module. To that end, watershed segmentation is performed by applying a distance transform on the binary image of the clutter (figure 2.14 (c)) and from the distance image a non-branching skeleton is derived (figure 2.14 (d)). Superimposing the skeleton on the clutter helps to extract the maxima. Passing from one maximum to the next detects the minimum where the cells should be separated. The separation produces a (filled) mask for each of the

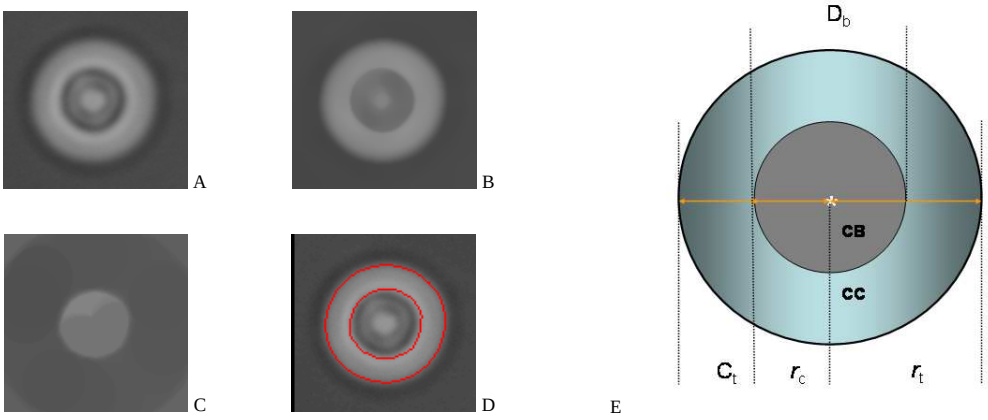
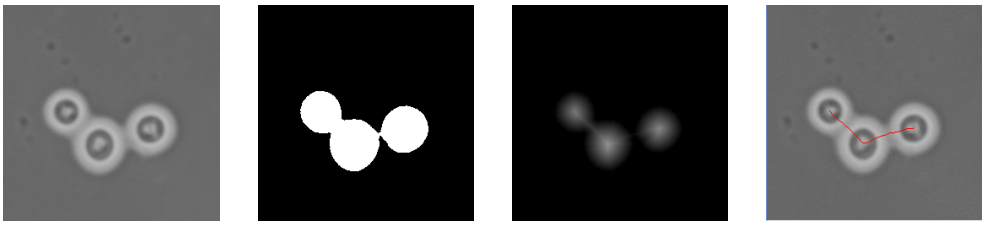


Figure 2.13: Imaging of the yeast cell with nigrosine background staining and showing obvious features from the cell. Figure A (panel, top left), depicts one cell with a dark background. The complete cell consists of the capsule and the cell body. The capsule is less light dense and thus using bright-field microscopy appears as white in the image of a yeast cell. The cell depicted has a relatively large capsule. The next three panels illustrate the steps to segmentation and the result. Figure B shows the result after filtering and enhancing the capsule so that through segmentation the total cell can be extracted. Figure C shows the result of filtering and enhancing the cell body of the yeast cell. Segmentation and XOR on the binary images produces the required result of a separate measure for the cell body and the capsule. Figure D shows the result as superimposed on the original image. Figure E (left panel), depicts a model of the yeast cell with a capsule. Using this schematic drawing the initial features for the recognition of the pathogen yeast cell can be understood. These features are closest to the recognition of the biologist. Thickness of capsule and cell radius are illustrated clearly. D_b = diameter cell body, C_t = capsule thickness (in pixel units), r_c = radius of cell body, r_t = radius of total cell. *CB*= Cell body, *CC* = Capsule. *CB* and *CC* are also used for surface area of cell body and capsule respectively.



A B C D

Figure 2.14: The processing pipeline for the segmentation and segmentation of cluttered cells. Figure A: a group of touching yeast cells as found in figure 2.12 A. Figure B: the result of the segmentation process. The labeled component would not adhere to the heuristic of circularity and hence it would be considered a cluster that needs to be separated. Figure C: the distance transform of 3B showing distinct maxima at the centroids of the individual cells. Figure D: the skeleton, derived from the distance transform and superimposed on the original image (black line).

cells in the clutter and the process continues with labeling (figure 2.11). The correctly separated cells are now processed by the segmentation module and prepared for measurement.

The individual cells that are extracted are used to determine the training/test set for classification. There is a range of features that can be measured from the shape of the cell and more specifically the capsule of the cell. In routine practice, the yeast biologist will evaluate capsule thickness through the microscope and possibly relate that to the radius of the cell. From digitized images, however, many more features can be derived. We will apply the measurements on the results from the segmentation procedure which are: a buffer image with the density image of one cell (for instance figure 2.13 (a)) and two masks, i.e., one for the cell body (CB) and one for the capsule (CC).

Feature Extraction

Initial analysis is directed to the features the biologist will check when examining a sample of a yeast culture. As indicated in the introduction, the thickness of the capsule may be an indication of the level of virulence of a particular yeast isolate. Therefore, this feature needs to be analyzed in a reproducible manner. As capsule size is easily made objective by digital measurement, relative measures are computed comparable to what the yeast biologist does routinely by investigating yeast cells by bright field microscopy using negatively stained cells. In figure 2.13 (e) a schematic drawing is given to illustrate the measurement of the capsule thickness.

We have taken the image moments as a starting point of our analysis and as the objects are relatively simple shapes these moments provide sufficient information for a classification on the basis of shape. From the image moments a set of features is derived that is used in the classifications and clustering.

In a two-dimensional density image, the image moments are expressed as:

$$m_{pq} = \iint x^p y^q f(x, y) dx dy \tag{2.4}$$

where $p + q$ indicates the order of the moment. For the case of a sampled image of size $N \times M$ this translates to:

$$M_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} x^p y^q f(x, y) \tag{2.5}$$

One should realize that in case of images with binary objects, the function $f(x, y)$ filters all irrelevant image information. In the case of binary images the image moments provide information on the geometrical distribution of a point set. The moments are made translation invariant by centering on the mean of the distribution. The mean is computed from the zero and first order moments. The centralized moments are expressed as:

$$\mu_{pq} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} (x - \bar{x})^p (y - \bar{y})^q f(x, y) \tag{2.6}$$

The first order moment in a binary image equals the object area; so the μ_{00} of CB is the area (in pixels) taken by CB and μ_{00} of CC is the area of the capsule. The sum of these zero order moments is the total area of the cell and thus we can express the relative area of CB and CC in terms of zero order moments as:

$$A_{CB}^{rel} = \frac{\mu_{00}^{CB}}{\mu_{00}^{CB} + \mu_{00}^{CC}} \tag{2.7a}$$

$$A_{CC}^{rel} = \frac{\mu_{00}^{CC}}{\mu_{00}^{CB} + \mu_{00}^{CC}} \tag{2.7b}$$

Instead of deriving r_t and r_c (cf. 2.13 (e)) from the area by using the analytical equation of a circle we use the data to find radii in the shape. These are the semi-major and semi-minor axis of the distribution, also known as the moments of inertia. The semi-minor and -major axes are computed (Verbeek 1995) (Verbeek 1999) from the

centralized second order moments as:

$$\alpha = \sqrt{\frac{2(\mu_{20} + \mu_{02} + \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})}{\mu_{00}}} \tag{2.8a}$$

$$\beta = \sqrt{\frac{2(\mu_{20} + \mu_{02} - \sqrt{(\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2})}{\mu_{00}}} \tag{2.8b}$$

In the same manner the relative thickness is derived from the semi-major and semi-minor axis:

$$T_{CC}^{rel-major} = \frac{\alpha^{cc}}{\alpha^{CB} + \alpha^{CC}} \tag{2.9a}$$

$$T_{CC}^{rel-minor} = \frac{\beta^{cc}}{\beta^{CB} + \beta^{CC}} \tag{2.9b}$$

Besides the major and minor axis we can derive a radius of gyration in both the x direction, the y direction and additional unified form:

$$\gamma_x^{CC} = \sqrt{\frac{\mu_{20}}{\mu_{00}^2}} \tag{2.10a}$$

$$\gamma_y^{CC} = \sqrt{\frac{\mu_{02}}{\mu_{00}^2}} \tag{2.10b}$$

$$\gamma_{xy}^{CC} = \sqrt{\frac{\mu_{20} + \mu_{02}}{\mu_{00}^2}} \tag{2.10c}$$

The third order moments relate to the skewness of the distribution in both the x and y direction as follows (Verbeek 1995):

$$Sk_x^{CC} = \frac{\mu_{30}}{\sqrt[3]{\mu_{20}^2}} \tag{2.11a}$$

$$Sk_y^{CC} = \frac{\mu_{03}}{\sqrt[3]{\mu_{02}^2}} \tag{2.11b}$$

The kurtosis (peakedness) is derived from the fourth order moments in both the x

and y direction as follows:

$$K_x^{CC} = \frac{\mu_{40}}{\mu_{20}^2} - 3 \tag{2.12a}$$

$$K_y^{CC} = \frac{\mu_{04}}{\mu_{02}^2} - 3 \tag{2.12b}$$

From the moments a set of 7 invariants can be derived (Hu 1962), (Gonzales & Woods 1993). These invariants are computed through a normalization of the centralized moments. We use the first four invariants; we will express the invariants in terms of normalized moments without further addressing the normalization step.

The first and second invariant are derived from second order moments; the third and fourth invariants are computed from third order moments:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{2.13}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11} \tag{2.14}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{2.15}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{2.16}$$

In our experiments the moments are computed for both the binary image and the gray-value images. The binary images are the cell body (*CB*) and the cell capsule (*CC*), both available from the segmentation. In addition, the binary images are used to mask out the gray-values under the area of *CB* and *CC* respectively, so that these can be used to compute a gray-value moment set.

The equations 2.4-2.16 provide a lot of features that require computation of the centralized moment-set (eq. 2.6), thus the moment sets are first transposed to the centralized form. This is accomplished for *CB* in binary and gray-value images as well as *CC* in binary and gray-value images. The binary measurements are related to the geometrical distribution of the shape whereas the gray-value measurements, in the way applied in our experiments, are related to the density distribution under the shape. All features are derived from the centralized moments and formulated for the *CC*; in all cases the feature is mutatis mutandis derived for the *CB*.

In addition, the relative area and relative thickness have been introduced to rule out the effect of the size of the cell. The relative area (eqs. 2.7) is derived from the zero order moment (in the binary case equal to area) and the relative thickness (eqs. 2.9) is derived from the semi-major and the semi-minor axis (cf. eqs. 2.8).

2.3.3 Experiments

Clustering and classification is carried out using the WEKA data mining toolkit which incorporates a wide variety of pattern recognition methodologies. Different

methods are easily compared and data import is dealt with through standard file formats. Feature selection, clustering and classification modules from WEKA were used to complete this study (Witten & Frank 2000).

As stated before, the goal from a biological perspective is to classify yeast cells into potentially virulent and non-virulent classes. We approximated this objective by building classifiers that can distinguish between different classes of yeast cells that resemble either virulent or non-virulent classes, with respectively thick or thin capsules. The image data set that we have used to conduct our experiments is sufficient in a proof of concept setting.

A wide variety of clustering and classification models have been built to investigate the usefulness of the various features generated by the image preprocessing phase, ranging from features that measure capsule size directly to the more abstract moment invariants. For completeness we also experimented with various types of classification algorithms, though we feel that at this stage this is not a major factor in the quality of the final classifier.

A major point we would like to mention, is that distinguishing between images may not be the same as distinguishing between categories of yeast cells. For instance, differences can be caused by varying environmental conditions when growing the cells, or can result from staining, image acquisition, feature extraction and other preprocessing steps. It is crucial to control these conditions as much as possible, which is only achievable to a certain extent. The rise of collaborative web collections of images may make matters rather worse than better, as image production becomes more separated from image distribution and analysis, and collections become heterogeneous. So, in addition, it is essential to focus on feature extractors that extract the right type of information, i.e. focusing on cell characteristics, and this was a core issue we have kept in mind for our entire approach.

Three sets of images were available: a Cryptococcus neoformans strain with thick capsules (7936), a Cryptococcus neoformans mutant with thin capsules (7926) and a related strain, namely Cryptococcus gattii (6955) also with a thick capsule. This allows us to zoom in on detecting cell characteristics that are typically associated with virulence. The third set of images comes from a different strain (6955) which allowed us to check the performance of classifiers if more classes are introduced. We performed a univariate analysis to estimate the predictive power of each attribute, using the information gain measure. To explore the usefulness of the set of attributes from a multivariate point of view we carried out a clustering on the data sets and evaluated the mapping of classes to resulted clusters. We then created an array of classification models using the procedure outlined below. All these experiments were carried out for the two-class (7926 vs. 7936) and three-class problem (7926, 7936, 6955).

As discussed the image preprocessing procedure produces an instance for each cell, resulting in a number of instances for each of the classes 7926, 7936 and 6955.

A number of features produced by the preprocessing procedure were deemed either irrelevant for classification (f.e. x,y position), not invariant (f.e. size, area and radius) or prone to measuring differences in imaging or preprocessing conditions rather than cell characteristics (f.e. distributional characteristics of the inner part of the cell; same for binary images of the capsule). The remaining set of attributes, the base set, included gyration (x, y, unified for binary and gray values, cf., eqs. 2.10); variance, skewness (cf. eqs. 2.11) and kurtosis (cf. eqs. 2.12) of the distribution of the capsule; the first and second moment invariants (cf. eqs. 2.13, 2.14; binary and gray); surface inner area compared to entire cell (cf. eqs 2.7, binary and gray) and thickness of capsule compared to entire cell (cf. eqs. 2.9; binary). To investigate the contribution of various attributes, classification models were built on ten different subsets of attributes, see the results section for details.

As discussed both clustering and classification experiments were carried out. For the clustering experiments we used standard k-means clustering with two, respectively, three clusters and created a matrix to compare the distribution of classes over these clusters. The classification algorithms applied were decision stumps (split on a single attribute), J48/C45 decision trees, naive Bayes, 1-nearest neighbor and 5-nearest neighbor. For the latter three algorithms we investigated two variants, one using all attributes available, and one based on selecting the most important attributes first. Attribute selection was performed on train sets only using the correlation based feature subset method (CFS) with best first forward search (Hall & Holmes 2003), (Hall 1999), (Witten & Frank 2000). Classifiers were evaluated based on ten runs of tenfold cross validation for each classification algorithm - attribute set combination.

2.3.4 Results

Below we present the results of our experiments. Without sufficient results in the image processing modules no classification would be possible, therefore, we first summarize the results obtained through preprocessing and segmentation. A number of clustering and classification experiments are summarized in tables 2.4 to 2.7; the specific focus of the classifications and the content of the tables is discussed in the second part of this section.

Preprocessing and segmentation

For the experiments described in total 84 images were processed. The images contained fully separated as well as cluttered cells. For the CBS 6955 cells 18 images were processed and 75 cells were extracted, for CBS 7926 cells 30 images were processed and 136 cells were extracted, whereas for CBS 7936 cells 36 images were processed and 66 cells were extracted. The difference in numbers in the CBS 7936 and CBS 7926 is caused by the fact that CBS 7926 is a mutant strain with significantly smaller size

Two class problem		
Attribute	**Eq.**	**Info Gain**
ϕ_1 CC-g	2.13	0.91
ϕ_1 CB-b	2.13	0.91
ϕ_2 CB-g	2.14	0.91
rel. area CC-g	2.7b	0.87
ϕ_1 CC-b	2.13	0.87
rel. area CC-b	2.7b	0.87
rel. thickness α CC-b	2.9a	0.87
rel. thickness β CC-b	2.9b	0.87
kurtosis y CC-g	2.12b	0.80
kurtosis x CC-g	2.12a	0.75
ϕ_3 CB-g	2.15	0.70
ϕ_4 CB-g	2.16	0.66
gyration ratio CC-b	2.10c	0.56
gyration x CC-b	2.10a	0.56
gyration y CC-b	2.10b	0.56

Three class problem		
Attribute	**Eq.**	**Info Gain**
ϕ_1 CB-g	2.13	1.23
ϕ_1 CC-g	2.13	1.00
ϕ_2 CB-g	2.14	1.00
ϕ_1 CC-b	2.13	0.97
rel. area CB-b	2.7a	0.97
rel. thickness α CC-b	2.9a	0.97
rel. thickness β CC-b	2.9b	0.97
rel. area CC-g	2.7b	0.94
kurtosis y CC-g	2.12b	0.86
ϕ_3 CB-g	2.15	0.81
ϕ_4 CB-g	2.16	0.77
kurtosis x CC-g	2.12a	0.68
skewness y CC-g	2.11b	0.56
gyration y CB-g	2.10b	0.55
gyration ratio CC-b	2.10c	0.52

Table 2.4: Predictive power of the attributes for the two-class (7936; 7936) and three-class (6955; 7936; 7926) classification tasks, measured in information gain on the full data set (15 top predictors out of 50).

as they practically do not have a capsule; apparently more of these cells were present in one sample.

For each of the image sets (6955, 7926, and 7936) the results of cells that were successfully segmented and measured were saved to flat files. These files were imported in the classification environment. The files contain a large number of features some of which are not relevant for the classifications we have pursued in our experiments. A selection was made on the basis of information gain measurements of the features.

Clustering and classification results

Now we present the results for the various data mining experiments. First, we assessed the predictive power of individual attributes by calculating the information gain over the full training data (Witten & Frank 2000), see table 2.4 (15 top predictors out of 50). The abbreviations of figure 2.13 E are used to indicate the measurement at hand; a suffix *b* is added if it concerns measurement of a binary object, whereas a suffix of *g* is added if the measurement concerns a gray-value object. The corresponding equation number is given in the second column. It is interesting to note the dominance of grayscale over binary image attributes. There is no clear winner between inner area and capsule attributes. The first and second moments invariants (equations 2.13, 2.14) dominate the top predictors.

The results of the clustering experiments can be found in table 2.5. If clusters

	Cluster 1	Cluster 2
7926	0	136
7936	66	0

	Cluster 1	Cluster 2	Cluster 3
7926	130	0	6
7936	0	64	2
6955	0	28	47

	Cluster 1	Cluster 2
7926	134	2
7936	0	66
6955	0	75

Table 2.5: Allocation of classes over clusters for two class – two cluster, three class – three cluster and three class – two cluster experiments respectively.

emerge that have a natural mapping to classes, it provides evidence that a good set of attributes is used to separate the classes. Note that this is a sufficient, not a necessary condition: in theory it is unlikely, but still possible, that classes are easily separable, but distributed over a multitude of clusters. However in our case there is a very good mapping from clusters to classes. It is also interesting to note that if we use two clusters on the three class problem, both classes with relatively thick cells (7936 and 6955) are grouped into a single cluster.

Finally, classifiers were built for the two-class and three-class classification problem, see table 2.6 and table 2.7 for the results. Classifiers were built using different combinations of attributes and classifiers; the base set of attributes is described in section 2.6. The attribute set was varied across using binary (description ends with *b*), gray level (ends with *g*) or both binary and gray value features (ends in *all*). Furthermore, we differentiate between using all base features, relative area only, relative thickness only and moments only.

It is quite clear from the two class results that image classes seem to be perfectly separable. As highlighted before, this does not guarantee that we can perfectly distinguish between the two classes of cells because in principle there could be other causes for differences between images. That said, from visual inspection it is clear that both classes of cells are quite different. Furthermore, the cluster experiments have shown that the two most similar classes end up in a single cluster when forced by the clustering algorithm (i.e. 7936, 6955). Note furthermore that a simple single split is sufficient for good performance (decision stump) and that this result is robust across the various sets of attributes. It is actually quite common in both very hard

	Decision Stump		J48		Naive Bayes		1-NN		5-NN	
Attribute Set	%	σ	%	σ	%	σ	%	σ	%	σ
Base set, all	99.4	1.9	99.0	2.3	99.5	1.5	100.0	0.0	100.0	0.0
Base set, b	98.5	2.5	99.0	2.3	99.5	1.5	100.0	0.0	100.0	0.0
Eq. 2.4 (CB, CC),b	98.5	2.5	99.0	2.3	97.8	2.9	98.5	2.5	98.6	2.8
Eq. 2.6 (CC),b	98.5	2.5	99.0	2.1	98.0	2.9	98.3	2.7	98.6	2.8
Eq. 2.10-13, b	98.5	2.5	99.0	2.3	98.9	2.3	99.5	1.6	99.5	1.5
Base set, g	100.0	0.0	99.4	1.9	100.0	0.0	100.0	0.0	100.0	0.0
Eq. 2.4 (CB, CC),g	99.4	1.6	98.9	2.3	98.6	2.5	99.0	2.1	99.2	1.9
Eq. 2.6 (CC),g	98.5	2.5	98.9	2.5	98.0	2.9	98.5	2.5	98.5	3.0
Eq. 2.10-13, g	100.0	0.0	99.4	1.9	100.0	0.0	100.0	0.5	100.0	0.5
Eq. 2.10-13, all	99.4	1.9	99.0	2.3	99.5	1.5	100.0	0.0	100.0	0.5

	Naive Bayes		1-NN		5-NN	
Attribute Set	%	σ	%	σ	%	σ
Base set, all	99.7	1.2	100.0	0.0	100.0	0.0
Base set, b	100.0	0.0	100.0	0.0	100.0	0.0
Eq. 2.4 (CB, CC),b	97.8	2.9	98.5	2.5	98.6	2.8
Eq. 2.6 (CC),b	98.0	2.9	98.3	2.7	98.6	2.8
Eq. 2.10-13, b	100.0	0.5	100.0	0.0	99.0	2.0
Base set, g	100.0	0.0	100.0	0.0	100.0	0.0
Eq. 2.4 (CB, CC),g	98.6	2.5	99.0	2.1	99.2	1.9
Eq. 2.6 (CC),g	98.0	2.9	98.5	2.5	98.5	3.0
Eq. 2.10-13, g	100.0	0.0	100.0	0.5	100.0	0.5
Eq. 2.10-13, all	100.0	0.0	100.0	0.5	100.0	0.5

Table 2.6: Classification accuracy and standard deviation (tenfold ten runs) for various combinations of attributes and classifiers (two class problem). Top table shows results without and bottom table with attribute selection.

	Decision Stump		J48		Naive Bayes		1-NN		5-NN	
Attribute Set	%	σ	%	σ	%	σ	%	σ	%	σ
Base set, all	76.2	1.7	95.1	3.7	91.9	4.8	96.2	3.4	94.5	4.0
Base set, b	74.3	2.9	91.7	5.0	87.6	5.6	91.8	4.5	92.6	4.7
Eq. 2.4 (CB, CC),b	74.3	2.9	73.8	3.6	75.0	5.6	82.1	5.8	78.3	6.6
Eq. 2.6 (CC),b	74.3	2.9	73.6	3.5	73.0	5.6	78.1	6.4	78.2	6.4
Eq. 2.10-13, b	74.5	2.8	81.7	5.9	83.1	5.9	81.7	5.8	81.3	6.1
Base set, g	76.2	1.7	94.9	3.8	94.6	4.4	97.3	3.0	95.2	3.7
Eq. 2.4 (CB, CC),g	72.9	3.6	84.0	5.7	79.6	6.5	79.7	6.8	83.6	6.1
Eq. 2.6 (CC),g	74.5	2.8	73.4	3.5	73.4	5.4	77.3	7.4	77.6	6.6
Eq. 2.10-13, g	76.2	1.7	88.1	4.9	87.3	5.4	83.8	6.1	81.3	5.8
Eq. 2.10-13, all	76.2	1.7	90.1	5.7	84.0	5.6	81.2	6.6	84.6	6.0

	Naive Bayes		1-NN		5-NN	
Attribute Set	%	σ	%	σ	%	σ
Base set, all	93.4	4.9	97.0	3.3	96.1	3.6
Base set, b	88.2	4.9	93.0	4.3	94.1	3.8
Eq. 2.4 (CB, CC),b	75.0	5.6	82.1	5.8	78.3	6.6
Eq. 2.6 (CC),b	73.0	5.6	78.1	6.4	78.2	6.4
Eq. 2.10-13, b	81.6	5.9	82.0	6.3	82.9	6.3
Base set, g	96.2	3.8	97.6	2.7	96.7	3.0
Eq. 2.4 (CB, CC),g	79.6	6.5	79.7	6.8	83.6	6.1
Eq. 2.6 (CC),g	73.4	5.4	77.3	7.4	77.6	6.6
Eq. 2.10-13, g	87.3	5.4	83.8	6.1	81.3	5.8
Eq. 2.10-13, all	82.2	6.0	83.6	5.3	84.4	5.8

Table 2.7: Classification accuracy and standard deviation (tenfold ten runs) for various combinations of attributes and classifiers (three class problem). Top table shows results without and bottom table with attribute selection.

and very easy real world data mining problems that simple models produce accurate and robust results (Holte 1993), (van der Putten & van Someren 2004).

To make the classification a bit less trivial we have also built classifiers to separate all three classes. As can be seen from table 2.7 the classification accuracy goes down for many attribute set - classification algorithm combinations. However, for some it is still possible to get near perfect results. It is interesting to note that the classifiers on moment invariants generally perform better than classifiers built on metrics that more or less directly aim to measure capsule thickness (relative area of the inner part; thickness of the capsule). This demonstrates that there is more to a cell than just the capsule thickness.

Furthermore, grayscale features seem to outperform features derived from binary images. This suggests there is more to a cell image than a binary shape. The relatively high performance of 1 nearest neighbor (1-NN) is also noteworthy. We do not have a definitive explanation for this, however, 1-NN tends to perform well if there are 'exceptions to the rule' that are actually not outliers, but valid examples of a class.

2.3.5 Discussion and Conclusion

We have described the development of a classification system for a capsulated pathogenic yeast. Apart from being successful in its own right, it serves as an example to other systems in the ingredients that are used and the emphasis on each of the parts that make up the system: i.e., the specimen preparation, the image acquisition, the image processing and analysis and the classification including feature selection.

The outcome for the different features with respect to the classification is intriguing. If we go by the judgment of the yeast biologist we would have to let the capsule features relating to geometry dominate. This is, however, not unambiguously found in the feature selection. Rather than binary features, the first and second moment invariants are shown to be important in the classifications. One should realize that these features were derived from a masked gray-value image. For these features the density distribution is only considered at the masked geometry, i.e., *CB* and *CC*. The precise mechanisms that make features discriminative can't be concluded from these experiments. New, controlled, experiments should be designed to get further insight in these features.

The three class experiments provide interesting findings. The third strain (6955) corresponds to the C. neoformans stain (7936) in that it has clearly visible (thick) capsules; genetically, however, these strains are different. The imaging conditions of 7926 and 7936 were more or less similar; the 6955 strain was captured under different conditions. One could be tempted to conclude that the results for the three clusters and the classification experiments may be weakened by this fact, since the classification and clustering models could be focusing on imaging conditions rather than yeast cell differences. Yet, we observe that when the three classes are forced into two clusters, the 6955 class is correctly grouped with the 7936 class, which indicates that the models and underlying attributes are detecting cell differences, not just differences in imaging conditions. On the basis of these experiments we can't draw definitive conclusions on the influence of the imaging conditions. It would, however, certainly be interesting to further investigate which of the features are truly invariant.

In our experiments we have controlled for all parameters other than the particular yeast strain used (yeast strain production, staining, image acquisition, automated segmentation and feature analysis etc.). This allows us to conclude us with appropriate level of confidence that with our methods we are not just differentiating between images, but also between the yeast strains. That said, large historical image collections are generally not produced under controlled conditions. An important next step would be to make the classification problem more complex by adding more noise, using images that have been captured under a wide variety of conditions. This will require more robust classifiers and methodologies to ensure that true differences between classes are learned and not just 'accidental' differences between images.

The results presented are based on bright field images of negatively stained specimen preparations. Recently, antibodies against the cryptococcal capsule have become available. If these antibodies are tagged with fluorescent dyes, the acquisition can be done based of fluorescence microscopy and consequently the segmentation procedure can be further improved and simplified. Moreover, instead of the 2D approach that has been applied in our experiments, Confocal Laser Scanning Microscopy (CLSM) could be used to be able to include 3D-features. The issue of separation of cells can, in that case, be solved by other means. The use of 3D–images will, however, require using other features, as the approach taken can't be translated directly from the 2D-case.

We have focused on the image features to discriminate between potentially virulent and non-virulent cells. From the point of view of content-based multimedia retrieval we will be moving in the direction of solutions where the yeast biologist actively includes more and different data in the analysis. This requires that researchers are able to have a lot of different search, navigation and browsing dimensions to access the data. Some of these will be lower level, syntactic, but others will be more high level semantic categories (like virulent/non virulent) that have an important connotation to the biologist. Prior to the analysis, data should be submitted to a database that will incorporate direct links to the relevant bio-molecular repositories (Bei, Belmamoune & Verbeek 2006). With respect to classifiers this will dictate these to be built for a wide array of classes. Ideally, automated classification procedures can be developed so that an end user, like a biologist, instead of a data miner, can create and train classifiers (see also the next case).

The tool presented here will allow automated analysis of capsular characteristics of many cryptococcal cells of isolates of different phenotypic or genetic background. This will be particularly useful when gene knock out strains of C. neoformans are being prepared and need to be analyzed for pathogenicity-related features. As stated earlier, the capsule is one of the most important characteristics to that respect. Furthermore, automated feature extraction and comparison of capsular characteristics will allow integrative studies where capsular characteristics are being compared with other features, which may be either phenotypic or genetic in nature. Among these are rates of melanization, expression profiles of virulence-related proteins, growth rates at different temperatures and substrates, assimilation patterns of carbohydrates, nitrogen compounds or vitamins, susceptibility to antifungals, genotypic data on the various subtypes known to exist in the species, and more importantly, extensive collections of transcriptome data as revealed by microarray analysis.

In conclusion, we have presented a holistic overview of an end to end process for classifying yeast cells using image features with the ultimate goal to detect pathogen conditions. Previous studies were based on manual measurement of capsule thickness and cell area in the binary image, but no automated procedures existed.

By carefully controlling the conditions we were able to show that through a

largely automated procedure we could distinguish between a yeast strain and its mutant (7936, 7926), which simulate respectively pathogenic or non-pathogenic cells. We have shown that there are more predictive features than simply thickness and area in the binary image, some related to the density distribution in the image, or under the shape of interest, in particular the first and second moment invariants. Furthermore, we have shown that when we introduce noise in the form of a third class of a distinctly different strain this is clustered in the proper class and classifiers that need to distinguish between the three classes still achieve acceptable accuracy.

We have identified various application scenarios and associated challenges for extending the solution, such as identifying more semantically interesting classes beyond pathogenicity, making classifiers more robust for heterogeneous sets of images and developing methodologies enabling biologists rather than data miners to develop classification modules for the purpose of content based image classification, annotation and retrieval.

2.4 Video Classification by End Users

This fourth case introduces a real time automatic scene classifier for content-based video retrieval. In our envisioned approach end users such as television archive documentalists, not image processing experts, build classifiers interactively, by simply indicating positive examples of a scene. Classification consists of a two stage procedure. First, small image fragments called patches are classified into building block classes (e.g., buildings, water, grass, crowd, skin). Second, frequency vectors of these patch classifications are fed into a second classifier for global scene classification (e.g., city, inside/outside, countryside). The first stage classifiers can be seen as a set of highly specialized, trained feature detectors, as an alternative to letting an image processing expert determine abstract features a priori. The end user or domain expert thus builds a visual alphabet that can be used to describe the video footage in features that are relevant for the task at hand. We present results for experiments on a variety of patch and image classes. The scene classifier approach has been successfully applied to other domains of video content analysis, such as content based video retrieval in television archives, automated sewer inspection, and porn filtering.

In our opinion in most circumstances the Holy Grail in video classification and in data mining in general, would be to let end users create classifiers, primarily because it will be more scalable; more classifiers can be created in shorter time by more people. Also, end users are problem domain experts and may use their knowledge to choose the right semantic features to recognize top level scenes, thus creating niche specific classifiers that can beat purely data and abstract feature driven approaches (van der Putten 1999*e*), (Israel et al. 2004*a*), (Israël et al. 2004*b*), (Israël et al. 2006).

2.4.1 Introduction

This work has been done as part of the EU Vicar project (IST). The aim of this project was to develop a real time automated video indexing, classification, annotation, and retrieval system. Vicar was developed in close cooperation with leading German, Austrian, Swedish, and Dutch broadcasting companies. These companies generally store millions of hours of video material in their archives. To increase sales and reuse of this material, efficient and effective video search with optimal hit rates is essential. Outside the archive but inside a broadcasting environment, large amounts of video material are managed as well, such as news feeds and raw footage, materials that an archive traditionally not even has access to (van der Putten 1999*e*), (Israël et al. 2004*b*).

Generally, only a fraction of the content is annotated manually and these descriptions are typically rather compact. Any system to support video search must be able to index, classify, and annotate the material extensively, so that efficient mining and search may be conducted using the index rather than the video itself. Furthermore, these indices, classifications, and annotations must abstract from the pure syntactical appearance of the video pixels to capture the semantics of what the video is about (e.g. as an idealized example, a description such as 'a shot of Obama jogging in a park').

Within Vicar a variety of visual events is recognized, including shots, camera motion, person motion, persons, and faces, specific objects, etc. In this chapter we will focus on the automated classification of visual scenes. For searching and browsing video scenes, classifiers that extract the background setting in which events take place are a key component. Examples of scenes are indoor, outdoor, day, night, countryside, city, demonstration, and so on. The amount of classes to be learned is generally quite large – tens to hundreds – and not known beforehand. So, it is generally not feasible to let an image processing expert build a special purpose classifier for each class.

Using our envisioned approach, an end user like an archive documentalist or a video editor can build classifiers by simply showing positive examples of a specific scene category. In addition, an end user may also construct classifiers for small image fragments to simplify the detection of high level global scenes, again just by showing examples (e.g., trees, buildings, and road).

We call these image fragments patches. The patch classifiers actually provide the input for the classification of the scene as a whole. The patch classifiers can be seen as automatically trained data preprocessors generating semantically rich features, highly relevant to the global scenes to be classified, as an alternative to an image processing expert selecting a set of abstract features e.g., wavelets, Fourier transforms). This removes the dependency on having an image processing expert available and as such decreases the time to market for new classifiers to be built and drastically increases the volume of classifiers that can be made available. But whilst volume of models and time to market are key benefits, there could also be

a quality benefit. The interactive procedure is a way to exploit a priori knowledge the documentalist may have about the real world, rather than relying on a purely data driven or abstract image processing approach. In essence, the end user builds a visual alphabet that can be used to describe the world in terms that matter to the task at hand, which may result in models that beat more traditional classifiers.

Note that the scene is classified without relying on explicit object recognition. This is important because a usable indexing system should run at least an order of magnitude faster than real time, whereas object recognition is computationally intensive. More fundamentally, we believe that certain classes of semantically rich information can be perceived directly from the video stream rather than indirectly by building on a large number of lower levels of slowly increasing complexity. This position is inspired by Gibson's ideas on direct perception (Gibson 1979). Gibson claims that even simple animals may be able to pick up niche specific and complex observations (e.g., prey or predator) directly from the input without going through several indirect stages of abstract processing.

This section is expository and meant to give a non-technical introduction into our methodology. A high level overview of our approach is given in section 2.4.2, and sections 2.4.3 and 2.4.4 discuss related work and positioning of our method. Section 2.4.5 provides more detail on the low level color and texture features used and section 2.4.6 specifies the classification algorithms used. Experimental results for patch and scene classification are given in section 2.4.6 and are discussed in section 2.4.7. We then describe three applications in which scene classification technology has been embedded (section 2.4.8). We finish with a conclusion (section 2.4.9).

2.4.2 Approach

In Vicar a separate module is responsible for detecting the breaks between shots. Then for each shot a small number of representative key frames is extracted, thus generating a storyboard of the video. These frames (or a small section of video around these key frames) are input to the scene classifier.

The scene classifier essentially follows a two stage procedure: (i) Small image segments are classified into patch categories (e.g., trees, buildings, and road) and (ii) these classifications are used to classify the scene of the picture as a whole (e.g., interior, street and forest). The patch classes that are recognized can be seen as an alphabet of basic visual elements to describe the picture as a whole.

In more detail, first a high level segmentation of the image takes place. This could be some intelligent procedure recognizing arbitrarily shaped segments, but for our purposes we simply divide images up into a regular n by m grid, say 3 by 2 grid segments for instance. Next, from each segment patches (i.e., groups of adjacent pixels within an image, described by a specific local pixel distribution, brightness, and color) are sampled. Again, some intelligent sampling mechanism

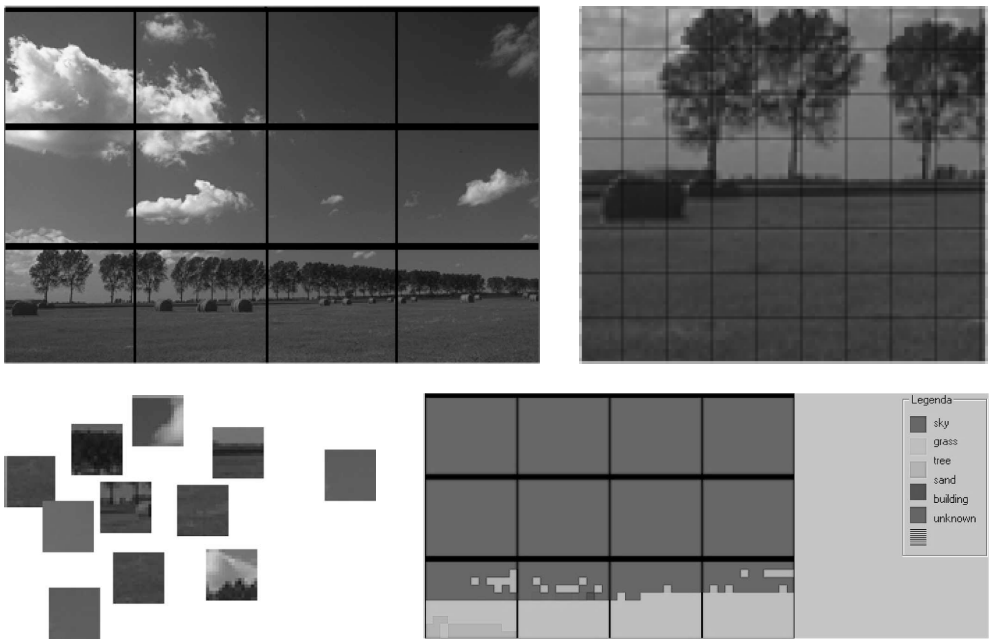


Figure 2.15: Screenshots visualizing the first phase of the scene classification process. From top to bottom, left to right: the images with a 4×3 grid over it, extraction of the patches from a grid cell, classification of the patches, and the resulting "patch image" with its legend.

could be used to recognize arbitrarily sized patches. However, we divided each grid segment by a second grid, into regular size image fragments, ignoring any partial patches sampled from the boundary. These patches are then classified into several patch categories, using color and texture features (see section 2.4.5). See figure 2.15, for a visualization of this approach. For each segment, a frequency vector of patch classifications is calculated. Then, these patch classification vectors are concatenated to preserve some of the global location information (e.g., sky above and grass below) and fed into the final scene classifier. Various classifiers have been used to classify the patches and the entire picture, including *k*-nearest neighbor and backpropagation neural networks.

2.4.3 Related Work

Literature on scene classification is relatively limited. Early retrieval systems like QBIC (Niblack, Barber, Equitz, Flickner, Glasman, Petkovic, Yanker & Faloutos 1993), (Flickner, Sawhney, Niblack, Ashley, Huang, Dom, Gorkani, Hafner, Lee, Petkovic, Steele & Yanker 1995), VisualSEEk (Smith & Chang 1997), PicHunter (Cox, Miller, Minka & Papathomas 2000), PicToSeek (Gevers & Smeulders 2000), and SIMPLIcity (Wang 2001) as well as more recent systems such as MARVEL (IBM Research 2005), M4ART (van den Broek, Kok, Schouten & Hoenkamp 2006), and the system proposed by Wu, Rahman & Chow (2005), use color, shape, and texture representations for picture search. Minka & Picard (1996), Picard (1995) and Picard & Minka (1995) extended Photobook with capabilities for classifying patches into so-called 'stuff' categories (e.g., grass, sky, sand, and stone), using a set of competing classification models (society of models approach). In Blobworld, pictures are segmented into regions with coherent texture and color of arbitrary shape ('blobs') and the user can search on specific blobs rather than the low level characteristics of the full picture (Belongie, Carson, Greenspan & Malik 1997), (Carson, Belongie, Greenspan & Malik 2002). However, these blobs are not classified into stuff nor scene categories (Belongie et al. 1997, Carson et al. 2002). Campbell et al also segment pictures into arbitrarily shaped regions and then use a neural network to classify the patches into stuff-like categories such as building, road, and vegetation (Campbell, Mackeown, Thomas & Troscianko 1996), (Campbell, Mackeown, Thomas & Troscianko 1997).

Some papers are available on classification of the scene of the picture as a whole. Lipson, Grimson & Sinha (1997) recognize a limited set of scenes (mountains, mountain lakes, waterfalls, and fields) by deriving the global scene configuration of a picture and matching it to a handcrafted model template. For example, the template for a snowy mountain states that the bottom range of a picture is dark, the middle range very light and the top range has medium luminance. Ratan & Grimson (1997) extend this work by learning the templates automatically. The templates are built using the dominant color-luminance combinations and their spatial relations in images of a specific scene category. They present results for fields and mountains only. Both papers only report results for retrieval tasks, not for classification.

Oliva & Torralba (2001) defined global characteristics (or semantic axes) of a scene (e.g., vertical – horizontal, open – closed, and natural – artificial), for discriminating between, for example, city scenes and nature scenes. These characteristics are used to organize and sort pictures rather than classify them. Gorkani & Picard (1994) classified city versus nature scenes. The algorithms used to extract the relevant features were specific for these scenes (i.e., global texture orientation). In addition, Szummer & Picard (1998) classified indoor and outdoor scenes. They first classified local segments as indoor or outdoor, and then classified the whole image as such. Both classifiers performed well, but it is not known whether these approaches generalize to other scene categories.

2.4.4 Positioning the Visual Alphabet Method

Our method uses the local patch classification as input for the classification of the scene as a whole. To our knowledge only Fung and Loe (Fung & Loe 1999*a*, Fung & Loe 1999*b*) reported a similar approach. Note that the final scene classifier only has access to patch class labels. From the point of view of the final classifier, the patch classifiers are feature extractors that supply semantically rich and relevant input rather than generic syntactic color and texture information. Moreover, the patch classifiers are trained rather than being feature extractors a priori selected by an image processing expert.

So, our method differs and improves on the general applicability for a variety of scene categories, without the need to select different and task specific feature extraction algorithms, for each classification task. Moreover, we used computationally cheap algorithms, enabling real time scene classification. A more fundamental difference is that we allow end users to add knowledge of the real world to the classification and retrieval engines, which means that it should be possible to outperform any purely data driven approach, even if it is based on optimal classifiers. This is important given the fact that image processing expertise is scarce and not available to end users, but knowledge of the world is abundant.

2.4.5 Patch Features

In this section, we discuss the patch features as used for patch classification. These provide the foundation for the scene classifier. In order of appearance, we discuss: (i) color quantization using a new distributed histogram technique, and histogram configurations (ii) human color categories, color spaces, and the segmentation of the HSI color space, and (iii) an algorithm used to determine the textural features used.

Distributed Color Histograms

At the core of many color matching algorithms lies a technique based on histogram matching. This is no different for the current scene classification system. Let us, therefore, define a color histogram of size n. Then, each pixel j present in an image, has to be assigned to a bin (or bucket) b. The bin b_i, with $i \in \{0, n-1\}$, for a pixel j with value x_j, is determined using:

$$\beta_i = \frac{x_j}{s} \tag{2.17}$$

where x_j is the value of pixel j and s is the size of the intervals, with s determined as follows:

$$s = \frac{\max(x) - \min(x)}{n} \tag{2.18}$$

with $\max(x)$ and $\min(x)$ respectively the maximum and minimum value x_j can take. For convenience, equation 2.18 is substituted into equation 2.17, which yields:

$$\beta_i = \frac{n \cdot x_j}{\max(x) - \min(x)} \tag{2.19}$$

Now, b_i is defined as the integer part of the decimal number β_i.

As for each conversion from an originally analog to a digital (discrete) representation, one has to determine the precision of the discretization and with that the position of the boundaries between different elements of the discrete representation. In order to cope with this problem, we distributed each pixel over three bins, instead of assigning it to one bin.

Let us consider an image with p pixels that has to be distributed over n bins. Further, we define $\min(b_i)$ and $\max(b_i)$ as the borders of bin i (b_i). Then, when considering an image pixel by pixel, the update of the histogram for each of these pixels, is done as follows:

$$b_i \quad += \quad 1 \tag{2.20}$$

$$b_{i-1} \quad += \quad 1 - \frac{|x_j - \min(b_i)|}{\max(b_i) - \min(b_i)} \tag{2.21}$$

$$b_{i+1} \quad += \quad 1 - \frac{|x_j - \max(b_i)|}{\max(b_i) - \min(b_i)} \tag{2.22}$$

where $\min(b_i) \leq x_j \leq \max(b_i)$, with $i \in \{0, n-1\}$ and $j \in \{0, p-1\}$. Please note that this approach can be applied on all histograms, but its effect becomes stronger with the decline in the number of bins a histogram consists of.

Histogram Configurations

Several histogram configurations have been presented over the years (van den Broek, van Rikxoort & Schouten 2005). For example, the PicHunter (Cox et al. 2000) image retrieval engine uses a HSV($4 \times 4 \times 4$) (i.e., 4 Hues, 4 Saturations, and 4 Values) quantization method. In Smith & Chang (1995) a HSV($18 \times 3 \times 3$) bin quantization scheme is described. The QBIC configuration used 4096 bins (Niblack et al. 1993), Flickner et al. (1995) uses RGB($16 \times 16 \times 16$). For more detailed discussions concerning color quantization we refer to Prasad, Gupta & Biswas (2001), Redfield, Nechyba, Harris & Arroyo (2001), Schettini, Ciocca & Zuffi (2001), van den Broek, van Rikxoort & Schouten (2005)

Histogram matching on a large number of bins, has a major advantage: Regardless of the color space used during the quantization process, the histogram matching will have a high precision. Disadvantages of our approach are its high computational

complexity and poor generalization. When a coarse color quantization is performed, these disadvantages can be solved. So, since the system should work real-time and the classifiers have to be able to generalize over images, a coarse color quantization is needed. However, to ensure an acceptable precision, it is key that human color perception is taken into account during quantization. The combination of color space and the histogram configuration is crucial for the acceptance of the results by the user.

Human Color Categories

Forsyth & Ponse (2002) state: "It is surprisingly difficult to predict what colors a human will see in a complex scene; this is one of the many difficulties that make it hard to produce really good color reproduction systems." From literature it is known that people use a limited set of color categories (Berlin & Kay 1969), (Goldstone 1995), (Kay 1999), (Roberson, Davies & Davidoff 2000), (Derefeldt, Swartling, Berggrund & Bodrogi 2004), (van den Broek & van Rikxoort 2005). A color category can be defined as a fuzzy notion of some set of colors. People use these categories when thinking of or speaking about colors or when they recall colors from memory. Research from various fields of science emphasizes the importance of focal colors in human color perception. The use of this knowledge may provide the means for bridging the semantic gap that exists in image and video classification.

No exact definition of the number exists nor is the exact content of human color categories known. However, all research mentions a limited number of color categories: ranging between 11 (Berlin & Kay 1969), (van den Broek, Kisters & Vuurpijl 2005) and 30 (Derefeldt & Swartling 1995), where most evidence is found for 11 color categories. We conducted some limited experiments with subjective categories (categories indicated by humans) but these did not give better results to 16 evenly distributed categories, so for simplicity we used this categorization. Now that we have defined a coarse 16 bin color histogram to define color with, we need a color space on which it can be applied.

Color Spaces

No color quantization can be done without a color representation. Color is mostly represented as tuples of (typically three) numbers, conform certain specifications (that we name a color space). One can describe color spaces using two important notions: perceptual uniformity and device dependency. Perceptually uniform means that two colors that are equally distant in the color space are perceptually equally distant. A color space is device dependent when the actual color displayed depends on the device used.

The RGB color space is the most used color space for computer graphics. It is device dependent and not perceptually uniform. The conversion from a RGB image to a gray value image simply takes the sum of the R, G and B values and divides the

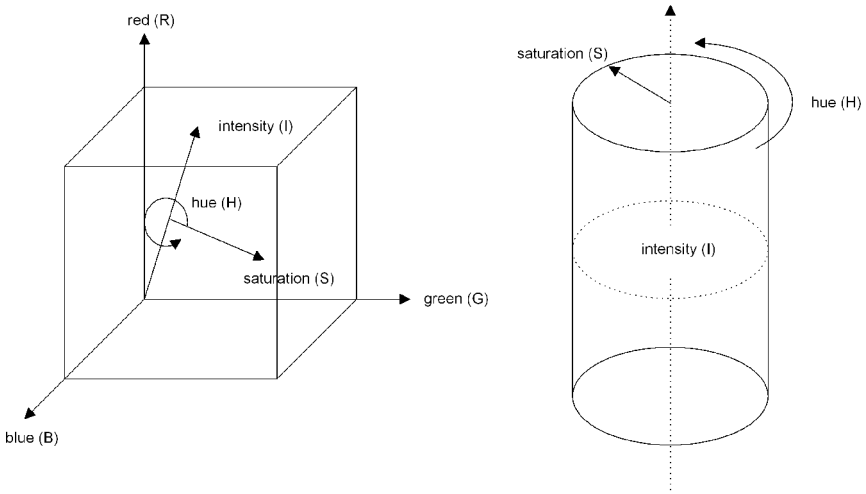


Figure 2.16: Left: The relation between the RGB and the HSI color space, from the perspective of the RGB color space. Right: The cylinder shaped representation of the HSI (hue, saturation, and intensity) color space, as used in this research.

result by three. The HSI / HSV (Hue, Saturation, and Intensity / Value) color spaces are more closely related to human color perception than the RGB color space, but are still not perceptual uniform. In addition, they are device-dependent. Hue is the color component of the HSI color space. When Saturation is set to 0, Hue is undefined and the Intensity / Value-axis represents the gray-scale image. Despite the fact that the HSI and HSV color spaces are not perceptually uniform, they are found to perform as good of better than perceptual uniform spaces such as CIE LUV (Lin & Zhang 2000). Therefore, we have chosen to use the HSI color space. Hereby, we took into account human perceptual limitations. If Saturation was below 0.2, Intensity was below 0.12, or Intensity was above 0.94, then the Hue value has not been taken into account, because for these Saturation and Intensity values the Hue is not visible as a color. Since image and video material is defined in the RGB color space, we needed to convert this color space to the HSI color space. This was done as follows (Gevers & Smeulders 1999):

$$H = \arctan\left(\frac{\frac{\sqrt{3}}{2}(G-B)}{R-\frac{1}{2}(G+B)}\right) \quad (2.23)$$

$$S = \sqrt{\left(R-\frac{\sqrt{3}}{2}(G-B)\right)^2+\left(\frac{1}{2}(G+B)\right)^2} \quad (2.24)$$

$$I = \frac{R+G+B}{3} \quad (2.25)$$

Note that, all H, S, and I values were normalized to values between 0 and 1.

Segmentation of the HSI Color Space

Our 16 color categories are defined by an equal division of the Hue axis of the HSI color space, since the Hue represents color. So far, only color was defined and luminance is ignored. Luminance is represented by the Intensity axis of the HSI color space. Again we have chosen for a coarse quantization: the Intensity-axis is divided into six equal segments. The original RGB color coordinates were converted to Hue and Intensity coordinates by equations 2.23 and 2.25, as adopted from Gevers & Smeulders (1999).

Next, for both the Hue and the Intensity histogram, using equation 2.19 each pixel is assigned to a bin. Finally equations 2.20, 2.21, 2.22 are applied on both histograms to update these. Since both histograms were a coarse quantization this method (i) is computationally cheap (making real-time classification possible) and (ii) facilitates in generalization by classifiers.

Texture

In addition to color, texture can be analyzed. Let us define texture as a repetitive arrangement of pixels values that either is perceived or can be described as such. For texture analysis, in most cases the Intensity of the pixels is used, hereby ignoring their color. Several techniques can be used to determine the patterns that may be perceived from the image (Rosenfeld 2001), (Palm 2004),(van den Broek & van Rikxoort 2005), (van Rikxoort, van den Broek & Schouten 2005), (van den Broek, van Rikxoort, Kok & Schouten 2006).

Most texture analysis methods derive textural features from the image, instead of describing arrangements of the individual pixels. This reduces the computational costs significantly, which is essential for applications working real time. Therefore, we used a texture algorithm that extracts three textual features for each position of a mask that is run over the image. Here, the size of the mask determines the ratio between local and global texture analysis. The position of the mask is defined by its

central pixel. Note that the mask is a square of $n \times n$ pixels, with n being an odd integer.

For each pixel of the mask, the difference between both its horizontal neighbors as well as the difference between its vertical neighbors is determined. (p, q) denotes the elements (i.e., pixels) of the image with (i, j) being the coordinates of the pixels located in a mask, surrounding an image pixel (p, q). Function f determines the normalized value of pixel (i, j) for a chosen color channel (i.e., H, S, or I), using equations 2.23, 2.24, and 2.25. Using the algorithm below, for each mask M_{11}, M_{12}, and M_{22} are determined, defining the symmetric covariance matrix M.

foreach$(p, q) \in$ Image
  foreach$(i, j) \in$ Mask(p, q)
    Sum + = $f(i, j)$
    SqSum + = $f(i, j)^2$
    M_{11} + = $(f(i + 1, j) - f(i - 1, j))^2$
    M_{12} + = $(f(i, j + 1) - f(i, j - 1))^2$
    M_{22} + = $(f(i + 1, j) - f(i - 1, j)) \cdot (f(i, j + 1) - f(i, j - 1))$

Let ev_1 and ev_2 be the eigenvalues of M. Given this algorithm, three textural features can be determined:

$$F_1 = \text{SqSum} - \text{Sum}^2 \tag{2.26}$$

$$F_2 = \frac{\min\{ev_1, ev_2\}}{\max\{ev_1, ev_2\}} \tag{2.27}$$

$$F_3 = \max\{ev_1, ev_2\} \tag{2.28}$$

F_1 (see equation 2.26) can be identified as the variance (σ^2), indicating the global amount of texture present in the image. The other two features, F_2 and F_3 (see equations 2.27 and 2.28), indicate the structure of the texture available.

If ev_1 and ev_2 differ significantly, stretched structures are present (e.g., lines). When ev_1 and ev_2 have a similar value (i.e., F_2 approximates 1; see equation 2.27), texture is isotropic. In the case both ev_1 and ev_2 are large (i.e., both F_2 and F_3 are large; see equation 2.27 and 2.28), clear structure is present, without a clear direction. In the case ev_1 and ev_2 are both small (i.e., F_2 is large and F_3 is small; see equation 2.27 and 2.28), smooth texture is present. Moreover, F_2 and F_3 are rotation-invariant.

Hence, this triplet of textural features provides a good indication for the textural properties of images, both locally and globally. In addition, it is computationally cheap and, therefore, very useful for real time content-based video retrieval. For more details, see for example Jähne (1997) on structure tensors. More recent work on nonlinear structure tensors has been presented by Brox, Weickert, Burgeth & Mrázek (2006).

2.4.6 Experiments and Results

In the previous section relevant features were introduced. These features were used for the first phase of classification: the classification of patches, resulting in a frequency vector of patch classes for each grid cell. In the second phase of classification, a classifier is used to classify the whole image. The input for the classifier is the concatenation of all frequency vectors of patch classes for each grid cell. So, two phases exist, each using their own classifier. We have experimented with two types of classifiers: A *k*-nearest neighbor classifier and a backpropagation neural network.

The advantage of *k*-nearest neighbor is that it is a lazy method, i.e. the models need no retraining. This is an important advantage given that we envisage an interactive learning application. However, given that *k*-nearest neighbor does not abstract a model from the data, it suffers more from the curse of dimensionality and will need more data to provide accurate and robust results. The neural network needs training, parameter optimization and performance tuning. However, it can provide good results on smaller data sets providing that the degrees of freedom in the model are properly controlled. The experiments all used a selection of images from the Corel image database as test bed.

Patch Classification

We will now discuss the patch classification experiments and results. Further below, the classification results of the image as a whole are discussed.

Each of the patches had to be classified to one of the nine patch categories defined (i.e., building, crowd, grass, road, sand, skin, sky, tree, and water). First, a *k*-nearest neighbor classifier was used for classification. This is because it is a generic classification method. In addition, it could indicate whether a more complex classification method would be needed. However, the classification performance was poor. Therefore, we have chosen to use a backpropagation neural network for the classification of the grid cells, with nine output nodes (as many as there were patch classes). For each of the nine patch classes both a train and a test set were randomly defined, with a size ranging from 950 to 2,500 patches per category. The neural network architecture was as follows: 25 input, 30 hidden, and 9 output nodes. The network ran 5,000 training cycles with a learning rate of 0.007.

With a patch size of 16×16, the patch classifier had an overall precision of 87.5%. The patch class crowd was confused with the patch class building in 5.19% of the cases. Sand and skin were also confused. Sand was classified as skin in 8.80% of the cases and skin was classified as sand in 7.16% of the cases. However, with a precision of 76.13% the patch class road appeared the hardest to classify. In the remaining 23.87% of the cases road was confused with one of the other eight patch classes, with percentages ranging from 1.55% to 5.81%. The complete results can be found in table 2.8.

Table 2.9 shows the results for a 8×8 patch classifier in one of our experiments. The 16×16 patch classifier clearly outperforms the 8×8 patch classifier with an overall precision of 87.5% versus 74.1%. So, the overall precision for the 8×8 patch classifier decreases with 13.4% compared to the precision of the 16×16 classifier. The decline in precision for each category, is as follows: sand 22.16%, water 21.26%, building 17.81%, skin 17.48%, crowd 17.44%, tree 16.8%, and road 7.16%. Only for the categories grass and sky the classification was similar for both patch sizes. Note that figure 2.15 presents a screenshot of the system, illustrating both the division of an image into grids. The classified patches are resembled by small squares in different grayscales.

So far, we have only discussed patch classification in general. However, it was applied on each grid cell separately: for each grid cell, each patch was classified to a patch category. Next, the frequency of occurrence of each patch class, for each grid cell, was determined. Hence, each grid cell could be represented as a frequency vector of the nine patch classes. This served as input for the next phase of processing: scene classification, as is discussed in the next subsection.

Scene Classification

The system had to be able to distinguish between eight categories of scenes, relevant for the Vicar project: interiors, city / street, forest, agriculture / countryside, desert, sea, portrait, and crowds. In pilot experiments several grid sizes were tested: a 3×2 grid gave the best results. The input of the classifiers were the normalized and concatenated grid vectors. The elements of each of these vectors represented the frequency of occurrence of each of the reference patches, as they were determined in the patch classification (see section 2.4.6).

Again, first a *k*-nearest neighbor classifier was used for classification. Similarly to the patch classification, the *k*-nearest neighbor classifier had a low precision. Therefore, we have chosen to use a neural network for the classification of the complete images, with eight output nodes (as many as there were scene classes). For each of the eight scene classes both a train and a test set were randomly defined. The train sets consisted of 199, 198, or 197 images. For all scene classes, the test sets consisted of 50 images. The neural network architecture was as follows: 63 input, 50 hidden, and 8 output nodes. The network ran 2,000 training cycles with a learning rate of 0.01.

The image classifier was able to classify 73.8% of the images correct. Interior (82%) was confused with city/street in 8.0% and with crowds in 6.0% of the cases. City/street was correctly classified in 70.0% of the cases and confused with interior (10%), with country (8.0%), and with crowds (6.0%). Forest (80%) was confused with sea (8.0%). Country was very often (28.0%) confused with forest and was sometimes confused with either city/street (6.0%) or desert (10%), which resulted in a low precision: 54.0%. In addition, also desert had a low precision of classification (64%); it was confused

	building	crowd	grass	road	sand	skin	sky	tree	water	unknown
building	**89.23**	3.02	0.09	1.11	1.02	0.60	0.38	3.70	0.85	0.00
crowd	5.19	**87.25**	0.19	1.81	0.44	0.50	0.38	2.94	0.06	1.25
grass	0.00	0.00	**94.73**	0.73	0.60	0.00	0.00	3.00	0.93	0.00
road	1.55	5.48	2.84	**76.13**	1.55	1.74	1.81	5.81	3.10	0.00
sand	1.84	0.88	2.24	1.44	**83.68**	8.80	0.24	0.00	0.64	0.24
skin	0.32	2.53	0.00	0.63	7.16	**89.37**	0.00	0.00	0.00	0.00
sky	0.21	0.00	0.00	2.57	0.93	0.00	**91.71**	0.36	3.86	0.36
tree	1.12	3.44	2.60	0.32	0.16	0.24	0.56	**88.44**	0.84	2.28
water	0.00	0.00	4.00	4.44	0.52	0.00	3.04	0.44	**87.26**	0.30

Table 2.8: Confusion matrix of the patch (size: 16×16) classification for the test set. The x-axis shows the actual category, the y-axis shows the predicted category.

	building	crowd	grass	road	sand	skin	sky	tree	water	unknown
building	**71.42**	9.00	0.85	2.69	2.43	2.86	0.26	6.53	0.77	3.20
crowd	10.38	**69.81**	1.13	1.56	2.13	5.56	0.69	6.44	0.19	2.13
grass	0.80	0.07	**93.87**	0.73	0.07	0.73	1.20	1.20	0.87	0.47
road	2.65	5.81	2.45	**68.97**	2.97	1.87	5.48	3.10	4.52	2.19
sand	3.44	3.12	2.88	1.84	**61.52**	15.20	8.80	0.16	2.80	0.24
skin	1.16	7.79	0.42	0.11	13.47	**71.89**	4.42	0.11	0.11	0.53
sky	0.00	0.00	0.00	0.29	1.36	2.57	**91.43**	0.07	4.07	0.21
tree	4.56	11.08	8.20	1.88	0.52	0.76	0.24	**71.64**	0.56	0.56
water	0.37	0.52	3.26	9.78	3.85	3.85	11.41	0.52	**66.00**	0.44

Table 2.9: Confusion matrix of the patch (size: 8×8) classification for the test set. The x-axis shows the actual category, the y-axis shows the predicted category.

with: interior (8.0%), city/street (6.0%), and with country (10%). Sea, portraits, and crowds had a classification precision of 80.0%. Sea was confused with city/street in 14%, portraits were confused with interior in 8.0% of the cases, and crowds were confused with city/street in 14.0% of the cases. In table 2.10 the complete results for each category separately are presented.

2.4.7 Discussion

Let us start with discussing the patch and scene classification results, before more moving on to more general topics.

For patch classification, two patch sizes have been applied. The 16×16 patch classifier gave clearly a much higher precision than the 8×8 patch classifier. Our explanation is that a 16×16 patch can contain more information of a (visual) category than a 8×8 patch. Therefore, some textures can't be described in a 8×8 patch (e.g., patches of buildings). A category such as grass, on the other hand, performed well with 8×8 patches. This is due to its high frequency of horizontal lines that fit in a 8×8 patch. Therefore, the final tests were carried out with the 16×16 patch size, resulting in an average result of 87,5% correct. Campbell and Picard reported similar results (Campbell et al. 1997), (Picard 1995), (Picard & Minka 1995) . However, our method

	Interior	City/street	Forest	Country	Desert	Sea	Portraits	Crowds
Interior	**82.0**	8.0	2.0	0.0	0.0	0.0	2.0	6.0
City/street	10.0	**70.0**	4.0	8.0	0.0	0.0	2.0	6.0
Forest	2.0	4.0	**80.0**	2.0	2.0	8.0	0.0	2.0
Country	0.0	6.0	28.0	**54.0**	10.0	0.0	0.0	2.0
Desert	8.0	6.0	2.0	10.0	**64.0**	4.0	4.0	2.0
Sea	4.0	14.0	0.0	2.0	0.0	**80.0**	0.0	0.0
Portraits	8.0	0.0	0.0	4.0	4.0	2.0	**80.0**	2.0
Crowds	4.0	14.0	0.0	0.0	2.0	0.0	0.0	**80.0**

Table 2.10: Confusion matrix of the scene classification for the test set. The x-axis shows the actual category, the y-axis shows the predicted category.

has advantages in terms of a much lower computational complexity. Moreover, the classified patches themselves are intermediate image representations and can be used for image classification, image segmentation as well as for image matching.

A challenge is the collection of training material for the patch classes to be recognized. Consequently, the development of an automatic scene classification system requires substantial effort since for all relevant patch classes, sets of reference patches have to be manually collected. For a given class, the other classes act as counterexamples. We are currently looking into several directions to reduce this burden. One approach would be to generate more counterexamples by combining existing patches. Another direction is the use of one class classification algorithms that only require positive examples (Tax 2001).

The second phase of the system consists of the classification of the image representation, using the concatenated frequency patch vectors of the grid cells. An average performance of 73.8% was achieved. The least performing class is country with 54% correct. What strikes immediately, when looking at the detailed results in table 2.9, is that this category is confused in 28% of cases with the category forest and in 10% of cases with the category desert. The latter confusions can be explained by the strong visual resemblance between the three categories, which is reflected in the corresponding image representations from these different categories. To solve such confusions, the number of patch categories could be increased. This would increase the discriminating power of the representations. Note that if a user searches on the index rather than on the class label, the search engine may very well be able to search on images that are a mix of multiple patches and scenes.

To make the system truly interactive, classifiers are needed that offer the flexibility of k-nearest neighbor (no or very simple training) but the accuracy of more complex techniques. We have experimented with learning algorithms such as naive Bayes, but the results have not been promising yet. Furthermore, one could exploit the interactivity of the system more, for instance by adding any misclassification identified by the user to the training data. Finally, the semantic indices are not only useful for search but may be used as input for other mining tasks. An example would be to use index clustering to support navigation through clusters of similar video material.

2.4.8 Applications

The scene classifier has been embedded into the VICAR system for content based video retrieval. In addition, the same visual alphabet approach has been used for other video classification applications such as porn filtering, sewage inspection and skin infection detection. The initial versions of these classifiers were built within very short time frames and with sufficient classification accuracy. This provides further evidence that our approach is a generally applicable method to quickly build robust domain specific classifiers. One of the reasons for its success in these areas, is its user-centered approach: the system can easily learn knowledge of the domain involved, by showing it new patch types and so creating a new visual alphabet, simply by selecting the relevant regions or areas in the image. In this section we will describe a number of these applications in a bit more detail.

Vicar Video Navigator

The scene classifier has been integrated into the Vicar Video Navigator. This system utilizes text-based search, either through manual annotations or through automatically generated classifications such as the global scene labels. As a result, Vicar returns the best matching key frames along with information about the associated video. In addition, a user can refine the search by combining a query by image with text-based search (van der Putten 1999*e*).

The query by image can either be carried out on local characteristics (appearance) or may include content based query by image. In the first case, the index consisting of the concatenated patch classification vectors is included in the search. In the latter case, the resulting index of scores on the global scene classifiers is used (content).

In figure 2.17 and 2.18, an example search is shown from a custom made web application based on the Vicar technology: the first screenshot shows one of the key frames that has been retrieved from the archive using the (automatically annotated) keyword countryside. An extra keyword person (also automatically annotated) is added in the search, as well as the content index of the image. In the second screenshot the results of the combined queries are shown: persons with a similar background scene as the query image.

Porn Filtering

To test the general applicability of our approach we built a new classifier to distinguish pornographic from non pornographic pictures. Within half a day a classifier was constructed with a precision of over 80%. As a follow up, a project for porn filtering was started within the EU Safer Internet Action Plan (IAP) program. Within this project, SCOFI, a real time classification system was built, which has been running in several schools in Greece, England, Germany and Iceland. The porn image

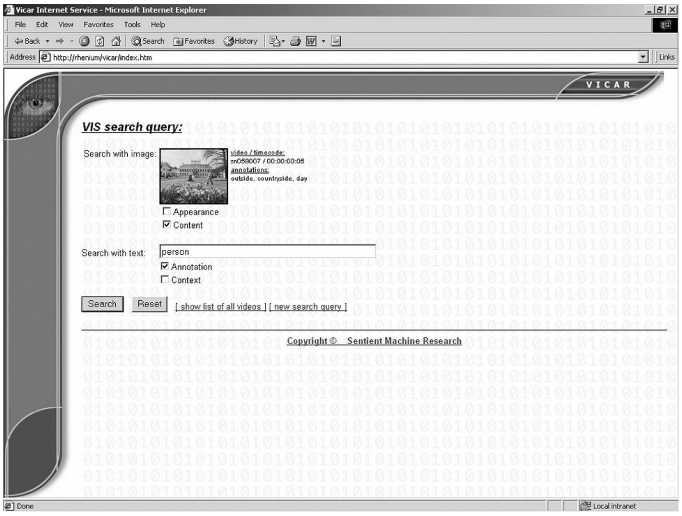


Figure 2.17: A query for video material.

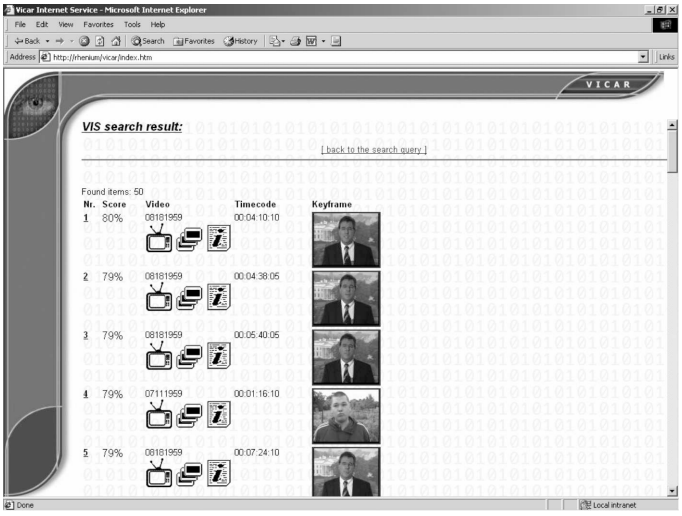


Figure 2.18: The result of a query for video material.

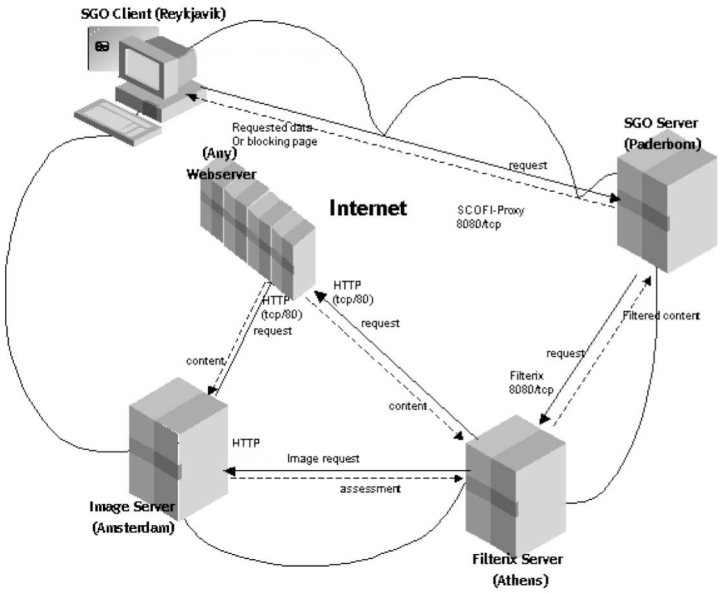


Figure 2.19: Different components of the SCOFI system: authentication server, text filtering server and porn image classification server

classifier is combined with a text classifier and integrated with a smart cards enabled authentication server to enable safe web surfing (see figure 2.19). The text classifier and the proxy server have been developed by Demokritos, Greece, and are part of the FilterX system (Chandrinos, Androutsopoulos, Paliouras & Spyropoulos 2000).

For this application of the system, we first created image representations using the patch classification network as mentioned in section 2.4.6. With these image representations we trained the second phase classifier, using 8000 positive (pornographic) and 8000 negative (non pornographic) examples. The results: the system was able to detect 92% of the pornographic images in a diverse image collection of 2000 positive examples and 2000 negative examples (which includes non pornographic pictures of people). There were 8% false positives (images that are not pornographic, are identified as pornographic images) and 8% false negatives. Examples of false positives were close ups of faces and pictures of deserts and fires. For a description of the complete results, we refer to Israël (1999). To improve results, within the SCOFI project a Vicar module was used that detects close ups of faces. The integrated SCOFI system that combines text and image classification has a performance of 0% overblocking

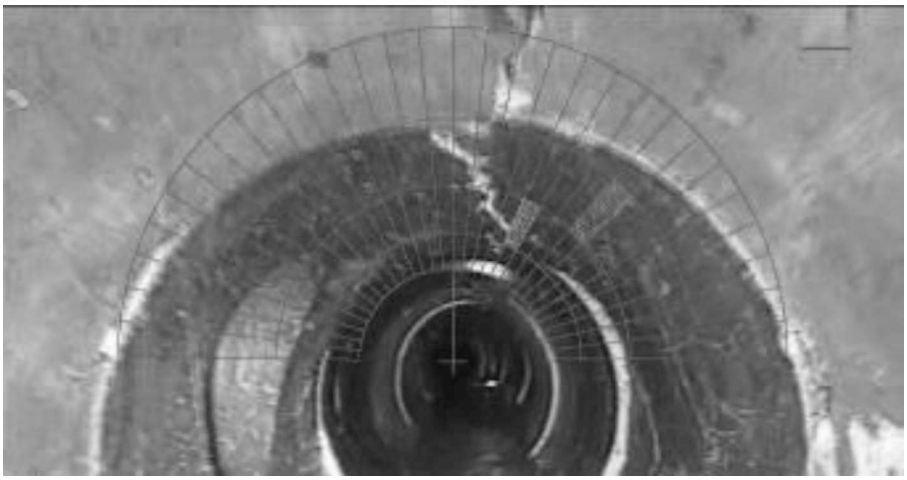


Figure 2.20: A spherical grid is placed on video footage of a sewer.

(i.e., 100% correct on non pornographic web pages) and 1% underblocking (i.e., 99% correct on pornographic web pages). As such it is used as a real time filter for filtering pornography on the Internet, in several schools throughout Europe.

Sewer Inspection

Our image classification approach is also applied to support the inspection of sewers in the RESEW project (EU GROWTH program for competitive and sustainable growth). Many European cities are spending increasing amounts to improve their sewage systems, so the inspection of deteriorating structures is becoming more and more important.

Currently, robots are used for sewer inspection, but these are completely controlled by operators and the video material that is collected is analyzed manually, which is a costly, time consuming and an error prone process. For instance, a UK based waste water utility company taking part in the project has 7,000 recent tapes of video material available, corresponding to thousands of kilometers of sewers. Real time, autonomous monitoring of the entire system would increase the need of automated analysis even further.

Automated and integrated systems for damage identification and structural assessment that are based on video analysis can be used to increase the speed and accuracy of the inspection and evaluation process and lower the cost. To prove the feasibility of the above the project partners have delivered an integrated and automated detection, classification, structural assessment and rehabilitation method selection system for sewers based on the processing of Closed Circuit Television

(CCTV) inspection tapes. The research prototype provides the user with an easy, fast and accurate method of sewer assessment. It consists of an intuitive interface to the sewage network with typical Geographic Information System functionality, a digital archive of indexed CCTV inspection tapes and a classification module to analyze video material for defects.

The RESEW classification method builds on the approach presented in this chapter. The primary goal of the classifier is to detect longitudinal cracks. First the central 'tunnel eye' is detected and a spherical rather than rectangular grid is placed around it (see figure 2.20; separate specialized modules extract the sewer joints and any CCTV text).

Neural networks are used to classify the extracted patches into crack and non crack classes. For this local patch classification we achieved an accuracy of 86.9%, with balanced train, validation and test sets of 40,000, 18,562 and 20,262 instances respectively. In the next stage, patch class histograms along the vanishing direction are classified to detect global longitudinal cracks. As an alternative method, a region growing approach is used that takes patch class probabilities as input. The latter approach generally produces more favorable results.

The environment is designed to be utilized in several utility contexts (water networks, sewer networks) where different engineering models are developed (e.g. structural reliability models for water pipes, reliability models taking into account seismic risk, safety models based on digital imagery of sewer interior, rehabilitation models for the previous). The system may be adapted to fit the needs of CCTV inspection of boreholes, shafts, gas and oil pipelines and other construction sectors. Going forward, the methods for analyzing the video material can also be used to build autonomous sewer robots that can explore sewage systems more or less independently.

2.4.9 Conclusion

In the work presented here, a general scene classifier is introduced that does not rely on computationally expensive object recognition. The features that provide the input for the final scene classification are generated by a set of patch classifiers that are learned rather than predefined, and specific for the scenes to be recognized rather than general.

Though the results on different scene categories can still be improved, the current system can successfully be applied as a generic methodology for creating domain specific image classifiers for content-based retrieval and filtering. This is demonstrated by its success in various applications such as the Vicar Video Navigator video search engine, the SCOFI real time filter for pornographic image material on the Internet and the RESEW sewer inspection system.

2.5 Lessons Learned

To summarize let us review some of the lessons learned in this chapter in relation to the thesis topics.

The first case is basically a non technical introduction in data mining, for an audience with no data mining or computer science background, in this example marketeers. This provides an inside out view of data mining, starting with the context rather than the technology. One of the findings of this study was that there were no clear winners in terms of prediction algorithms used across the overall range of problems. Only for specific problems, for specific data sets and only in the top segments, non linear algorithms such as neural networks were better able to 'cream the crop'. This is an indication for the need of evaluation methods that go beyond basic accuracy, and the need for methods to characterize a problem or algorithm rather than just assuming there is a silver bullet algorithm that will consistently beat the rest.

The second case is also example of an introduction of data mining for end users. This case appeared in a handbook for medical practitioners on head and neck cancer prognosis. The goal in this case is to predict five year survival probability for head and neck cancer patients. Evidence based medicine is becoming more important in the medical field, from empirically based studies towards medical decision support systems. As in the first case, for the specific data used in the experiments, the core modeling step is of lesser importance, as the top scoring models are relatively close in terms of performance. To explain the differences in classifiers we have performed a so called bias variance analysis. This demonstrates that for this problem the so called variance component of error is more important than bias for explaining the differences in performance across classification methods. This implies for instance that the emphasis should be on using simple, stable learners that base its parameter estimates on large number of instances rather than techniques that exploit complex, local relationships in data. A particular problem in this domain is that there are generally a number of data sets available from various hospitals or institutions, and each research group creates models on this data in isolation. So there is a need for procedures and end to end methodologies that combine or exploit data from various sources in a safe manner.

The third case discusses the classification of yeast cells to evaluate pathogen conditions. This case shows the full end to end process from growing yeast samples, capturing images, feature extraction, supervised and unsupervised data mining and evaluation of the results. Again for this problem we demonstrate that all classifiers perform roughly the same almost perfect performance. That said, it is still an open question whether the underlying problem is easy to solve (classifying yeasts) whereas the data mining problem is easy (classifying pictures). In our opinion this is a good example that in practice the particular translation of the research or business problem

into a data mining problem has a major impact on the results.

The fourth case introduces a real time automatic scene classifier for content based video retrieval. In our envisioned approach end users like documentalists, not image processing experts, build classifiers interactively, by simply indicating positive examples of a scene. To produce classifiers that are sufficiently reliable we have developed a procedure for generating problem specific data preprocessors. This approach has been successfully applied to other domains of video content analysis, such as content based video retrieval in television archives, automated sewer inspection, and porn filtering. In our opinion in most circumstances the ideal approach would be to let end users create classifiers, primarily because it will be more scalable; a lot more classifiers can be created in much shorter time.

In the remaining chapters we will address a number of specific research topics that are driven from applications similar to the ones above, but we aim to go beyond a single case or problem. One of the aims of this chapter was to show there is much more to data mining than just applying an algorithm to a data set. As a consequence, we feel that data mining research should not be limited to improving core modeling algorithms. Therefore the next chapters address specific topics concerning the end to end process (chapter 4), steps preceding modeling (the data step, chapter 3) and the steps following it (model evaluation and profiling, chapters 4 and 5).

Chapter 3

Data Fusion: More Data to Mine in

With no data, there is nothing to mine in. Multiple of sources of data can exist, and linking this data together can be non trivial. Assume we are given an instance, representing for example a customer or patient. The problem of merging information from different sources about this particular instance, assuming it can't be done with simple joins, is also called the exact matching problem (Radner, Rich, Gonzalez, Jabine & Muller 1980). Intelligent techniques are then used to determine what pieces of information from the various sources are concerned with a particular instance. In contrast, enriching the data for this instance with information from other instances is called a statistical matching or data fusion problem, which is the topic of this chapter. This can be seen as a form of data enrichment.

In literature data fusion is almost exclusively used in a market research or socio-economic survey context, to merge information from various samples with different sets of interview questions, in order to reduce the response burden or to connect survey data that has previously not been studied jointly. The resulting surveys are then typically mined using simple techniques such as cross tabulations and correlation analysis. However in data mining a more common task is prediction, so it is interesting to study whether it is possible to build better models by using data that has been enriched by data fusion. This is the topic of the research presented in this chapter, and it is based on papers that at the time of publication were the first to study it in this context (van der Putten 2000*a*), (van der Putten 2000*b*), (van der Putten, Kok & Gupta 2002*a*), (van der Putten, Kok & Gupta 2002*b*).

The goal of this research is not primarily to develop new algorithms, but to introduce data fusion to the data mining community by a proof of principle that demonstrates data fusion can add value for predictive modeling – which should not

be confused with a conclusion that it always will. As a leading example we focus on database marketing, however the results will generalize to any case where there is an interest to enrich data that will then be used for predictive modeling. The chapter also includes a short summary of a process model we developed for using data fusion in database marketing (van der Putten, Ramaekers, den Uyl & Kok 2002).

3.1 Introduction

Data mining papers often start with claiming that the exponential growth in the amount of data provides great opportunities for data mining. Reality can be different though. In real world applications, the number of sources over which this information is fragmented can grow at an even faster rate, resulting in barriers to widespread application of data mining and missed business opportunities. Let us illustrate this paradox with a motivating example from database marketing.

In marketing, direct forms of communication are becoming more popular. Instead of broadcasting a single message to all customers through traditional mass media such as television and print, customers receive personalized offers through the most appropriate channels, inbound (the customer contacts the company) and outbound (the company contacts the customer), in batch and real time. So it becomes more important to gather information about media consumption, attitudes, product propensity etc. at an individual level (van der Putten 1999*a*). Basic, company specific customer information resides in customer databases, but market survey data depicting a richer view of the customer are only available for a small sample of potentially anonymous customers.

Collecting all this information for the whole customer database in a single source survey would certainly be valuable, but prohibitively costly, if not impossible because of privacy constraints. The common alternative within business to consumer marketing is to buy syndicated socio-demographic data that have been aggregated at a geographical level. All customers living in a particular geographic location, for instance in the same zip code area, are associated with the same characteristics. This is also limited, given that in reality customers from the same area may behave differently. Furthermore, regional identifiers such as zip codes may be absent in company specific surveys because of privacy concerns.

The zip code based data enrichment procedure can be seen as a very crude example of data fusion: the combination of information from different sources. However more general and powerful fusion procedures are required that cater to any number and kind of 'linking' variables, without requiring a perfect match. Data mining algorithms can help to carry out such generalized fusions and create rich data sets for further data mining for marketing and other applications.

In this chapter we position data fusion as both an enabling technology and an interesting research topic for data mining and database marketing. A fair amount

of work has been done on data fusion over the past 30 years, but primarily outside the knowledge discovery and database marketing communities, as its application was primarily limited to media and socio-economic research. The wide majority of published cases we are aware of focus on fusing survey samples. However, our application domain of interest is database marketing, not market research. We are not so much interested in fusing surveys, but in enriching customer databases with market surveys to enable behavioral targeting for one to one marketing. To our knowledge we were the first to report on the added value of fusion for predictive analytics, by comparing models on data sets with and without fusion data (van der Putten 2000*a*), (van der Putten 2000*b*), (van der Putten, Kok & Gupta 2002*a*), (van der Putten, Kok & Gupta 2002*b*), (van der Putten, Ramaekers, den Uyl & Kok 2002). We are aware of only one recent study that discusses enriching customer databases with survey data for direct marketing purposes, which also refers to our publications this chapter was based on (van Hattum & Hoijtink 2008), (van Hattum & Hoijtink 2009).

Note that data fusion can act as an important enabler for data mining, but in return the data fusion problem can be seen as a data mining, intelligent systems or soft computing problem. In almost all published cases statistical matching is used which can be seen as a special case of *k*-nearest neighbor or fuzzy matching, but in principle any data mining technique could be applied. To conclude, data fusion is a fertile, new research area for data mining research, because it removes barriers for large scale data mining applications, and data mining techniques can be used for carrying out the fusion itself.

We would like to share and summarize the main approaches taken so far from a data mining perspective (section 2). A case study from database marketing serves as a clarifying example and a proof of principle result (section 3). We then generalize from the case results by giving a high level overview of a process model for carrying out data fusion projects for the purpose of mining customer databases (section 4). In section 5 we provide a summary and conclusions.

3.2 Data Fusion

Valuable work has been done on data fusion in areas other than data mining. From the end of the sixties until now, the subject has been both popular and controversial in economics and market research. Data fusion first emerged as a tool for economic policy research primarily in the U.S. and Germany, see for instance Budd (1971), Ruggles & Ruggles (1974), Barr & Turner (1978), Rodgers (1984), Little & Rubin (1986), Paass (1986), Rubin (1986), Kum & Masterson (2008); Radner et al. (1980) provides an overview. Later data fusion became quite a popular tool in media research to study the relationship between product usage and mass media consumption, with a focus on Europe and Australia. See for example O'Brien (1991), Roberts (1994), Jephcott & Bock (1998), Soong & de Montigny (2003), Soong & de Montigny (2004); Baker,

Harris & O'Brien (1989) provides an overview. See also Raessler (2002), D'Orazio, Zio & Scanu (2006) for statistically oriented textbooks.

Data fusion has yet to be discovered by the traditional knowledge discovery and machine learning communities as a standard topic for research, though a relatively new area is developing around mining uncertain data – note fused data can be seen as a special case of uncertain data (Pei, Getoor & de Keijzer 2009). Data fusion is also known as micro data set merging, statistical record linkage, multi-source imputation and ascription. Data fusion is sometimes used as a data mining related term in multi-sensor information fusion, however in that context it refers to a different concept: combining information from different sources about a single entity, where as in our case we enrich data about instance a (a customer for example) with information from other instances $b, c, \ldots$ (other customers).

Until today, in marketing data fusion is often used to reduce the required number of respondents or questions in a survey. For instance, for the Belgian National Readership survey questions regarding media and questions regarding products are collected in 2 separate groups of 10,000 respondents each, and then fused into a single survey, thereby reducing costs and the required time for each respondent to complete a survey. However, it is not commonly used yet to enrich customer databases.

3.2.1 Data Fusion Concepts

Let us introduce some key data fusion concepts. We assume that we start from two data sets. These can be seen as two tables in a database that may refer to disjoint data sets, i.e. it is actually not required that any of the instances in table 1 also occur in table 2. The data set that is to be extended is called the recipient set A and the data set from which this extra information has to come is called the donor set B. We assume that the data sets share a number of variables. These variables are called the common variables X. The data fusion procedure will add a number of variables to the recipient set. These added variables are called the fusion variables Z. Unique variables are variables that only occur in one of the two sets: Y for A and Z for B. See figure 3.1 for a marketing example. In general, we will learn a model for the fusion using the donor B with the common variables X as input and the fusion variables Z as output and then apply it to the recipient A.

3.2.2 Core Data Fusion Algorithms

In nearly all studies, statistical matching is used as the core fusion algorithm. The statistical matching approach can be compared to k-nearest neighbor prediction with the donor as training set and the recipient as a test or deployment set. The procedure consists of two steps. First, given some element from the recipient set, the set of k best matching donor elements is selected. The matching distance is calculated over the common variables, or a subset of these.

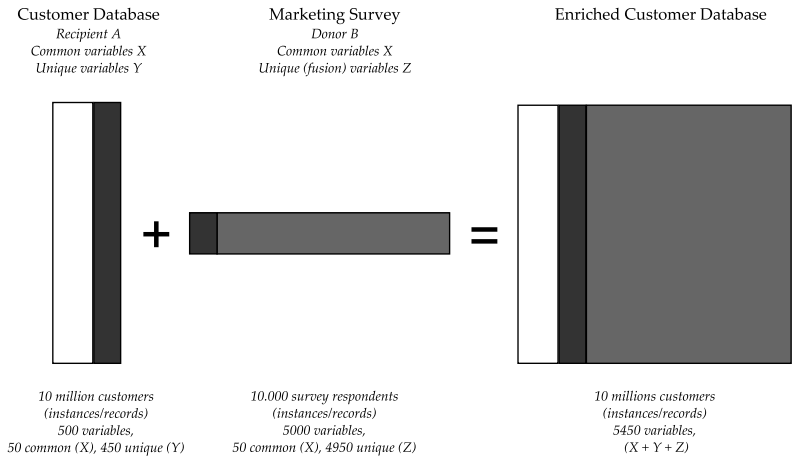


Figure 3.1: Data fusion for database marketing: a customer database is enriched with market survey information for further data mining

Standard distance measures such as Euclidian distance can be used, but often more complex measures are designed to tune the fusion process. For instance, it may be desirable that men are never matched with women, to prevent that 'female' characteristics such as 'pregnant last year' are predicted. In this case, the gender variable will become a so-called cell or critical variable; the match between recipient and donor must be 100% on the cell variable; otherwise these will not be matched at all. Weighting can be used to reflect the relative importance of each of the donor variables.

Another enhancement is called constrained matching. Experiments with statistical matching have shown that some donors are used more than others, even if the donor and recipient are large samples of the same population. This can result in a fusion that is not representative, as the values for the fusion variables for these donors have a larger influence on predictions. Especially for donors with an average profile this can be the case; this is an artifact of the winner takes all character of nearest neighbor combined with the fact that the signal can get lost in high dimensional, noisy data (regression to the mean). By taking into account how many times an element of the donor set has been used when calculating the distance, we can counter this effect (Barr & Turner 1978), (Rodgers 1984), (Baker et al. 1989), (van Pelt 2001), (Flores & Albacea 2007).

It is interesting to note that within data fusion research this is seen as a generally accepted problem, whereas within standard k-nearest neighbor research it is not identified as such. It may be that overusing donors is a problem, however it is not yet

proven whether penalizing donors makes things better or worse, especially because this can be hard to evaluate. This is an area that warrants more theoretical debate in our opinion. For a detailed discussion of some methods for constrained matching we developed and benchmarked see van Pelt (2001).

In the second step, the prediction for the fusion variables can be constructed using the set of best matching nearest neighbors, e.g. by calculating averages (numerical), modes (categorical) or distributions (categorical or numerical). In this step, the contribution of a neighbor is sometimes weighted inversely proportional to its distance from the recipient.

A number of constraints have to be satisfied by any fusion algorithm in order to produce valid results. Firstly, the donor must be representative for the recipient, or at least contain sub sets that are representative. This does not necessarily mean that the donor and recipient set need to be samples of the same population, although this would be preferable. For instance, in the case of statistical matching only the set of donors used in the fusion process needs to be representative of the recipient set. The recipients could be buyers of a specific product and the donor set could be very large sample of the general population that includes instances representative for these recipients. Methods that are not nearest neighbor based but that build a global, abstract model on the entire data set using donor data only, such as regression, may be more prone to errors in this example. This could be a possible explanation for the popularity of nearest neighbor based techniques for data fusion. Assuming the donor set is sufficiently large, the idea is that one can always find donors that are representative of the recipient, and predictions are made from these local recipient neighborhoods only ('product owners'). In contrast, a regression model to predict fusion variables would be developed on the donor data set, i.e. discover the relationships between common and fusion variables in the donor set alone ('the general population'), and the resulting global model would be applied to the recipient.

Secondly, the common variables must be good predictors for the fusion variables. In addition, the Conditional Independence Assumption must be satisfied: the commons X must explain all the relations that exist between unique variables Y and Z. In other words, we assume that $P(Y|X)$ is independent of $P(Z|X)$. This could be measured by the partial correlation $r(ZY, X)$, however if the recipient and donor data sets are disjoint there is no joint data available on X, Y and Z to compute this. As an intuitive explanation, consider there would be some other variable W that explains the relationship between Y and Z above and beyond what the commons X can explain; if it exists, finding out the exact relationship between Y and Z by predicting Z from X will not be possible. In most of our fusion projects we start with a small-scale fusion exercise to test the validity of the assumptions and to produce ballpark estimates of fusion performance.

In the wide majority of cases the standard statistical matching is being used, other approaches are quite rare. Below we will mention some of the key ones found,

from classical and genetic optimization, traditional statistics and data mining & soft computing.

In Barr & Turner (1978), constrained fusion is modeled as a large scale linear programming transportation model. The main idea was to minimize the match distance under the constraint that all donors should be used only once, given recipients and donors of equal size. This was recently extended to an approach that used genetic algorithms rather than classical optimization algorithms to solve the transportation problem (Flores & Albacea 2007); an alternative genetic algorithm approach to statistical matching can be found in Cubo, Robles, Segovia & Ruiz (2005). The (constrained) fusion problem can also be seen as a variant of the well-known stable marriage problem (Gusfield & Irving 1989) for which classical optimization solutions exist, some are briefly mentioned in (Baker et al. 1989).

In statistics extensive work has been done on dealing with missing data (Little & Rubin 1986), including likelihood based regression methods. Some researchers have proposed to impute values for the fusion variables using multiple models to reflect the uncertainty in the correct values to impute (Rubin 1986). In Kamakura & Wedel (1997) a statistical clustering approach to fusion is described based on mixture models and the Expectation Maximization (EM) algorithm. In van Hattum & Hoijtink (2008) and van Hattum & Hoijtink (2009) a comparison is made between logistic regression, nearest neighbor style statistical matching, latent cluster analysis and Bayesian model based clustering. See also Raessler (2002) and D'Orazio et al. (2006) for textbooks on statistical approaches to data fusion.

Traditional machine learning and data mining techniques are not yet often used for fusion. In Smith, Chuan & van der Putten (2001) also a clustering approach is taken, comparing *k*-means clustering with Self-Organizing Maps. Other data mining and soft computing techniques include the use of CART decision trees (Contrino, McGuckin & Banks 2000) and fuzzy logic (Noll 2009), (Noll & Alpar 2007). These examples of non nearest neighbor approaches are exceptions to the rule, and in most of the cases above only a single technique is being used. To address this gap we have executed fusion experiments comparing nearest neighbor based approaches with common data mining techniques such as naive Bayes, logistic regression, decision stumps, decision trees and feedforward neural networks (Maat 2006).

3.2.3 Data Fusion Evaluation and Deployment

An important issue in data fusion is to how to measure the quality of the fusion; this is not a trivial problem (Jephcott & Bock 1998). In our framework we distinguish between internal evaluation and external evaluation. This refers to the different steps in the data mining process. If one considers data fusion to be part of the data step and evaluates the quality of the fused data set only within this stage then this is an internal evaluation. However, if the quality is measured using the results within the

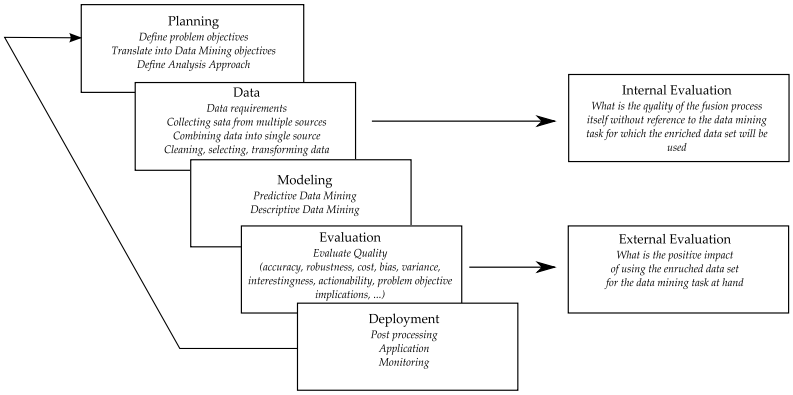


Figure 3.2: Internal and external evaluation of data fusion quality within the overall data mining process

other steps in the data mining process, then we call this an external evaluation (see figure 3.2).

Assume for instance that one wants to improve the response on mailings for a certain set of products, and this is the reason why the fusion variables would be added in the first place. In this case, one way to evaluate the external quality is to check whether an improved mail response prediction model can be built when fused data is included in the input.

Ideally, the fusion algorithm is tuned towards the kinds of analysis that is expected to be performed on the enriched data set. In practice the external evaluation will provide the bottom line evaluation, but an enriched data set could be used for multiple purposes unknown at the time of the fusion, and the internal evaluation will provide smoke test results about the fusion quality. In other words, a fusion that passes internal evaluation can still deliver bad external evaluation results, but a fusion with bad internal fusion results will likely not deliver good external test results.

3.3 Case Study: Cross Selling Credit Cards

As a case example of using data fusion for predictive data mining, assume the following example. A bank wants to learn more about its credit card customers and expand the market for this product. Unfortunately, there is no survey data available that includes credit card ownership; this variable is only known for customers in the customer base. Data fusion is used to enrich a customer database with survey data.

The resulting data set serves as a starting point for further data mining. The goal is to find out whether the enriched data has added value for the task at hand, i.e. predict who has a high probability to take up a credit card, and profile prospects in terms of survey variables, both of which can't be achieved using single source data only.

To simulate the bank case we do not use a separate donor; instead we draw a sample from an existing proprietary real world market survey (the Dutch SUMMO national readership survey) and split the sample into a disjoint donor set and recipient set, i.e. no donor instance can act as recipient and vice versa. The original survey contains over a 1000 variables and over 5000 possible variable values and covers a wide variety of consumer products and media. Whilst this is a simulation, it can be seen as representative for situations when the data sets to be fused are sufficiently large random samples from the same underlying population, which is a common use case especially in marketing.

Exceptions would be situations when samples differ by design or are poor samples of a population. An example of a difference by design is a customer database for a young and trendy mobile telecom provider versus survey on calling behavior for the general population in a given country. Note that some of the fusion methods presented in the previous section do not apply if samples are not meant to be representative, such as constrained matching. Another example of poor representativeness could be various small data sets on cancer patients for hospitals with different overall life expectancy rates.

The recipient set representing a small sample from the customer database, contains 2000 records with a cell variable for gender, common variables for age, marital status, region, number of persons in the household and income. Furthermore, the recipient set contains a unique variable for credit card ownership. One of the goals is to predict this variable for future customers. The donor set representing the survey contains the remaining 4880 records, with 36 variables for which we expect that there may be a relationship to the credit card ownership: general household demographics, holiday and leisure activities, financial product usage and personal attitudes. These 36 variables are either numerical or Boolean.

First we discuss the specific kind of matching between the databases and then the way the matching is transformed into values of the fusion variables. The matching is done on all common variables. Given an element of the recipient set we search for elements in the donor set that are similar. Elements of the donor set need to agree on the cell variable gender. All the common variables are transformed to numerical values and simple Euclidean distance on the commons is used as the distance measure. We select the k best matching elements from the donor. For the values of the fusion variables, we take the average of the corresponding values of the k best matching elements from the donor set. This statistical matching approach can be seen as k-nearest neighbor classification using the donor set as the search set, applied to the recipient set.

3.3.1 Internal evaluation

As a baseline analysis we first compared averages for all common variables between the donor and the recipient. As could be expected from the donor and recipient sizes and the fact that the split was done randomly, there were not many significant differences between donor set and recipient set for the common variables. Within the recipient 'not married' was over represented (30.0% instead of 26.6%), 'married and living together' was under represented (56.1% versus 60.0%) and the countryside and larger families were slightly over represented. This provides a baseline expectation of magnitude of differences that could be caused by sampling error only (or lack of representativeness by design if that would apply).

Then we compared the average values between the values of the fusion variables and the corresponding average values in the donor. Only the averages of 'Way Of Spending The Night during Summer Holiday' and 'Number Of Savings Accounts' differed significantly, respectively by 2.6% and 1.5%. Compared to the differences between the common variables, which were entirely due to sampling errors, this was a good result.

Next, we evaluated the preservation of relations between variables, for which we used the following measures. For each common variable, we listed the correlation with all fusion variables, both for the fused data set and for the donor. The mean difference between common-fusion correlations in the donor versus the fused data set was 0.12 ± 0.028. In other words, these correlations were well preserved. A similar procedure was carried out for correlations between the fusion variables with a similar result.

3.3.2 External evaluation

The external evaluation concerns the value of data fusion for further analysis. Typically only the enriched recipient database is available for this purpose. We first performed some descriptive data mining to discover relations between the target variable, credit card ownership, and the fusion variables using straightforward univariate techniques. We selected the top 10 fusion variables with the highest absolute correlations with the target (see table 3.1).

Note that this analysis was not possible without the fusion, because the credit card ownership variable was only available in the recipient. If other new variables become available for the recipient customer base, e.g. product ownership of some new product, their estimated relationships with the donor survey variables can directly be calculated, without the need to carry out a new survey.

We then investigated whether different predictive modeling methods would be able to exploit the added information in the fusion variables (the method for fusion, statistical matching, was not under investigation). The specific goal of the models was to predict a response score for credit card ownership for each recipient, so that these

Variable
Welfare class
Income household above average
Is a manager
Manages which number of people
Time per day of watching television
Eating out (privately): money per person
Frequency usage credit card
Frequency usage regular customer card
Statement current income
Spend more money on investments

Table 3.1: Top ten fusion variables in recipient most strongly correlated with credit card ownership

could be ranked from top prospects to suspects. We compared models trained only on values of common variables to models trained on values of common variables plus either all or a selection of correlated fusion variables. We used feed forward neural networks, linear regression, k nearest neighbor search and naive Bayes classification.

The feed forward neural networks had a fixed architecture of one hidden layer with 20 hidden nodes using a tanh activation function and an output layer with linear activation functions. The weights were initialized by Nguyen-Widrow initialization to enforce that the active regions of the layer's neurons were distributed roughly evenly over the input space (Nguyen & Widrow 1990). The inputs were linearly scaled between -1 and 1. The networks were trained using scaled conjugate gradient learning as provided within Matlab (Moller 1993). The training was stopped after the error on the validation set increased during five consecutive iterations. For the regression models we used standard least squares linear regression modeling. For the k nearest neighbor algorithm, we used the same simple approach as in the fusion procedure, so without normalization and variable weighting, with k=75. We used our own implementation of the standard Naive Bayes algorithm. The core fusion algorithm was implemented in C++ using an object oriented library we originally developed for codebook based algorithms (codebooks, Self Organizing Maps (SOM), LVQ etc. (van der Putten 1996)); the algorithms to build the prediction models were developed using MATLAB (de Ruiter 1999).

Error criteria such as the root mean squared error or accuracy do not always suffice to evaluate a ranking task. Take for instance a situation where there are few positive cases, say people that own a credit card. A model that predicts that no one is interested in credit cards has a low rmse, but is useless for the ranking and the selection of prospects. In fact, one has to take the costs and gains per mail piece into

Score list	**Corresponding *c*-index**
(0.1, 0.2, 0.3, 0.4, 0.5)	$\frac{1}{6} * ((3\frac{1}{2} + 4\frac{1}{2}) - 2) = 1$
(0.1, 0.2, 0.4, 0.3, 0.5)	$\frac{1}{6} * ((2\frac{1}{2} + 4\frac{1}{2}) - 2) = \frac{5}{6}$
(0.1, 0.2, 0.4, 0.4, 0.5)	$\frac{1}{6} * ((3 + 4\frac{1}{2}) - 2) = \frac{11}{12}$

Table 3.2: *C*-index calculation examples for the target list (0,0,0,1,1)

account. If we do not have this information, we can consider rank based tests that measure the concordance between the ordered lists of real and predicted cardholders.

We use a measure we call the *c*-index, which is a test related to Kendall's Tau (de Ruiter 1999). The *c*-index is a rank based test statistic that can be used to measure how concordant two series of values are, assuming that one series is real valued and the other series is binary valued.

We use the following procedure to calculate the *c*-index. Assume that all records are sorted ascending on rank scores. Records can be positive or negative (for example, if these are credit card holders or not). We assign points to all positive records: in fact we give $k - 0.5$ points to the k-th ranked positive record and records with equal scores share their points. These points are summed and scaled to obtain the *c*-index, so that an optimal predictor results in a *c*-index of 1 and a random predictor results in a *c*-index of 0.5. Under these assumptions, the *c*-index is equivalent (but not equal) to Kendalls Tau.

The scaling works as follows. Assume that l is the total number of points that we have assigned, and that we have a total of n records with s positive records. If the s positives all have a score higher than the other $n - s$ records, then the ranking is perfect and $l = s * (n - s/2)$. If the s positives all have a score that is lower than the $n - s$ others, then we have used a worst case model and $l = s^2/2$. The *c*-index is thus calculated by:

$$c - \text{index} = \frac{l - \frac{s^2}{2}}{s(n - \frac{s}{2}) - \frac{s^2}{2}} = \frac{l - \frac{s^2}{2}}{s(n - s)} \tag{3.1}$$

See table 3.2 for some examples. Note that by definition $c = 0.5$ corresponds to random prediction and $c = 1$ corresponds to perfect prediction.

We report results over ten runs with train and test sets of equal size. The results of our experiments can be found in table 3.3. We provide the average *c*-value and standard deviation over all runs. We also measure the statistical significance of im-

provements by fusion through a one tailed two sample T test. The p-value intuitively relates to the probability that the improvement gained by using fusion is coincidental.

The results show that for the data set under consideration most models that are allowed to take the fusion variables into account outperform the models without the fusion variables. Assuming variable selection, three results are significant at the 0.01 level, one at the 0.05 level and one is not significant (p=0.2). Without variable selection three results are significant at the 0.01 level, one is not significant (p=0.38) and one result is not better. So without significance testing, nine out of ten results are better, seven out of ten are better at the 0.05 significance level, and six out of ten results are better at the 0.01 level.

From an algorithm perspective, the best results are achieved with logistic regression using commons and correlated fusion variables. A possible explanation for this is that regression is a high bias method that can only model linear relationships. Fusion in this case may have added additional variables to the model that make the problem more linear (van der Putten & van Someren 2004). The results are significant with $p < 0.01$ for logistic regression, naive Bayes Gaussian and k-nearest neighbor. The neural network results are significant as well at the 0.05 level, provided variable selection has taken place, otherwise the results are not significant. This could be due to the fact that the number of degrees of freedom in a neural network, the network weights, is severely impacted by an increase in the number of inputs, so variable selection is even more important. The results for naive Bayes multinomial are actually worse if no variable selection has taken place, and with variable selection the improvement is not significant. This may be due to the fact that variables added are violating the naive Bayes assumption of independency, coupled with the issue of the multinomial over the Gaussian approach of having potentially too many unique values in the fusion variables to allow for proper estimation of model parameters.

For four out of five algorithms, using variable selection on the enriched data set improves the performance. Fusion variables are derived information, not measured and even if the fusion process were perfect, specific fusion variables may not be relevant for the prediction task at hand. Variable selection can successfully be used to counter this effect. Assuming variable selection, linear regression seems to benefit most from enrichment through fusion: a difference of 0.032 versus 0.019 (naive Bayes Gaussian), 0.013 (naive Bayes Multinomial) and 0.011 (neural networks).

In figure 3.3, cumulative response curves are shown for the linear regression models, for commons only and commons plus fusion variables. A response curve displays the probability of positive, in this case percentage of card holders (y-axis) for model selections of increasing size, ordered from top to bottom model score (x-axis). Response curves are often used in database marketing, for instance to compare model quality at a specific volume cut off of customers to be contacted (see also section 2.1.5). Curves for all the runs are displayed and logistic trend lines are fitted to the series for commons only and enriched data.

	Only common variables	**Common and correlated fusion variables**	**Common and all fusion variables**
SCG neural network	c=0.692 ± 0.012	c=0.703 ± 0.015 p=0.041	c=0.694 ± 0.019 p=0.38
Linear regression	c=0.692 ± 0.014	c=0.724 ± 0.012 p=0.000	c=0.713 ± 0.013 p=0.002
Naive Bayes Gaussian	c=0.701 ± 0.015	c=0.720 ± 0.012 p=0.003	c=0.719 ± 0.012 p=0.005
Naive Bayes multinomial	c=0.707 ± 0.015	c=0.720 ± 0.011 p=0.200	c=0.704 ± 0.009 p not relevant
***k*-nearest neighbor**	c=0.702 ± 0.012	c=0.716 ± 0.013 p=0.0093	c=0.720 ± 0.012 p=0.0023

Table 3.3: External evaluation results: using enriched data generally leads to improved performance.

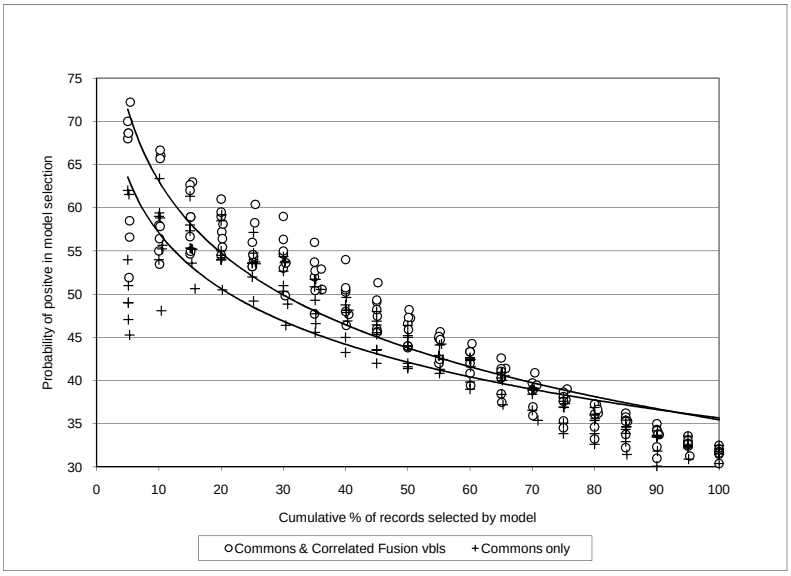


Figure 3.3: Cumulative response curves linear regression models for predicting credit card ownership (seven random runs) with and without fusion variables. The x-axis corresponds to the cumulative volume of top scoring instances selected by the model, the y-axis corresponds to the cumulative percentage of positives (cardholders) in the selection.

As can be seen from the graph at the 100% cut off, the overall percentage of credit card holders is 32.5%. In general credit card ownership can be predicted quite well: the top 10% of cardholder prospects according to the model contains around 50-65% cardholders, the top 20% contains 50-60% card holders still. The spread of results for smaller volumes is larger, this is common and due to a smaller sample size and hence a less robust estimation of the true percentage of cardholders in smaller selections. The added logarithmic trend lines clearly indicate that the models that include fusion variables are better in selecting the top prospects. At 10% the difference between trend lines is 6.0% (from 57.0% to 63.0% card owners), at 20% it is 4.1% (50.7% versus 54.8%), which is quite substantial and can translate to high impact on campaign ROI. For model selections over 40% the differences become a lot smaller. Again this is a common pattern; if the selection volume gets larger, the pool of cardholders to 'fish' from becomes substantially smaller, the overall percentage of cardholders drops, and the prediction task to select medium prospects is substantially noisier, so the various models will converge. From a business and customer centricity perspective these customers are less rewarding segments to contact in outbound campaigns, and in an inbound scenario medium or low propensity propositions will likely not 'win' over other propositions, so this section of the curve is generally of less interest.

3.3.3 Case Discussion

Data fusion can be a valuable tool for data mining practitioners. For descriptive data mining tasks, the additional fusion variables and the derived patterns may be more understandable and easier to interpret. This is not restricted to relations between commons and fusion variables, also relationships between variables that only appear in the recipient and donor can be studied, which can't be achieved without fusion if donor and recipient are disjoint data sets. An example would be profiling the users of a particular new product as indicated by the customer database in terms of answers to an older survey, without requiring that information about the product was actually part of the survey.

For predictive data mining, enriching a data set using fusion may make sense, notwithstanding the fact that the fusion variables are derived from information already contained in the donor variables. Fusion may make it easier for high bias algorithms such as linear regression to discover complex non-linear relations between commons and target variables by exploiting the information in the fusion variables. Of course, it is recommended to use appropriate variable selection techniques to remove the noise that is added by 'irrelevant' fusion variables and counter the 'curse of dimensionality', as demonstrated by the experiments (van der Putten & van Someren 2004).

There is also a practical dimension to this. Even if certain relationships could be studied by looking at single source data only for subsets of customers one would

need to have access and knowledge of these data sets, or knowledge how to combine the results of mining exercises on separate data sets into a single result. In many cases it can be more practical to let a core expert team fuse a variety of data sources into a single set on a periodical basis, and make this available to a wider community of customer insight analysts. This is valid not just for database marketing, but for instance also in the case of providing public integrated multi source data sets for scientific research, for instance in the medical domain (Maat 2006). In media research this is already common practice, as many national readership surveys are based on fused surveys. A central organization provides fused product usage and media consumption data, which is then used by media planning agencies, advertizers and publishers for media planning and target group profiling.

The fusion algorithms itself provide an interesting opportunity for further data mining research. There is no fundamental reason why the fusion algorithm should be based on *k*-nearest neighbor prediction instead of clustering methods, decision trees, regression, the expectation-maximization (EM) algorithm or other data mining algorithms, whereas examples are still rare. In addition, it is to be expected that future applications will require massive scalability. For instance, in the past the focus on fusion for marketing was on fusing surveys with surveys, each containing up to tens of thousands of respondents and hundreds of questions or more. In contrast, customer databases typically contain millions of customers. This requires scalable fusion algorithms, as well as scalable algorithms to mine the fused data, which also need to be able to deal with the uncertainty in this data.

It goes without saying that evaluating the quality of data fusion is also crucial. We hope to have demonstrated that this is not straightforward and that it ultimately depends on the type of data mining that will be performed on the enriched data set. As discussed, recently a new research area is developing around algorithms that are specifically adapted to mine uncertain data (Pei et al. 2009). Fused data sets can be seen as a special case of such data and the fusion process can actually generate metadata that provide an indication of the degree of uncertainty in the fused data. Explorative research has been carried out to see whether data mining algorithms that take data quality matrices into account can exploit this metadata (Sun 2005).

3.4 A Process Model for a Fusion Factory

In the previous sections we provided an example in which an enriched customer base provides an improved source for data mining, in this case better input to predict the propensity for a credit card. This provides proof of concept evidence for the feasibility of using data fusion for database marketing. However to take the step towards wide scale real world applications more is needed. This research project was carried out in the context of setting up a commercial data fusion service, a factory to carry out fusions on an ongoing, repeatable basis. So as a next step after proving the idea in

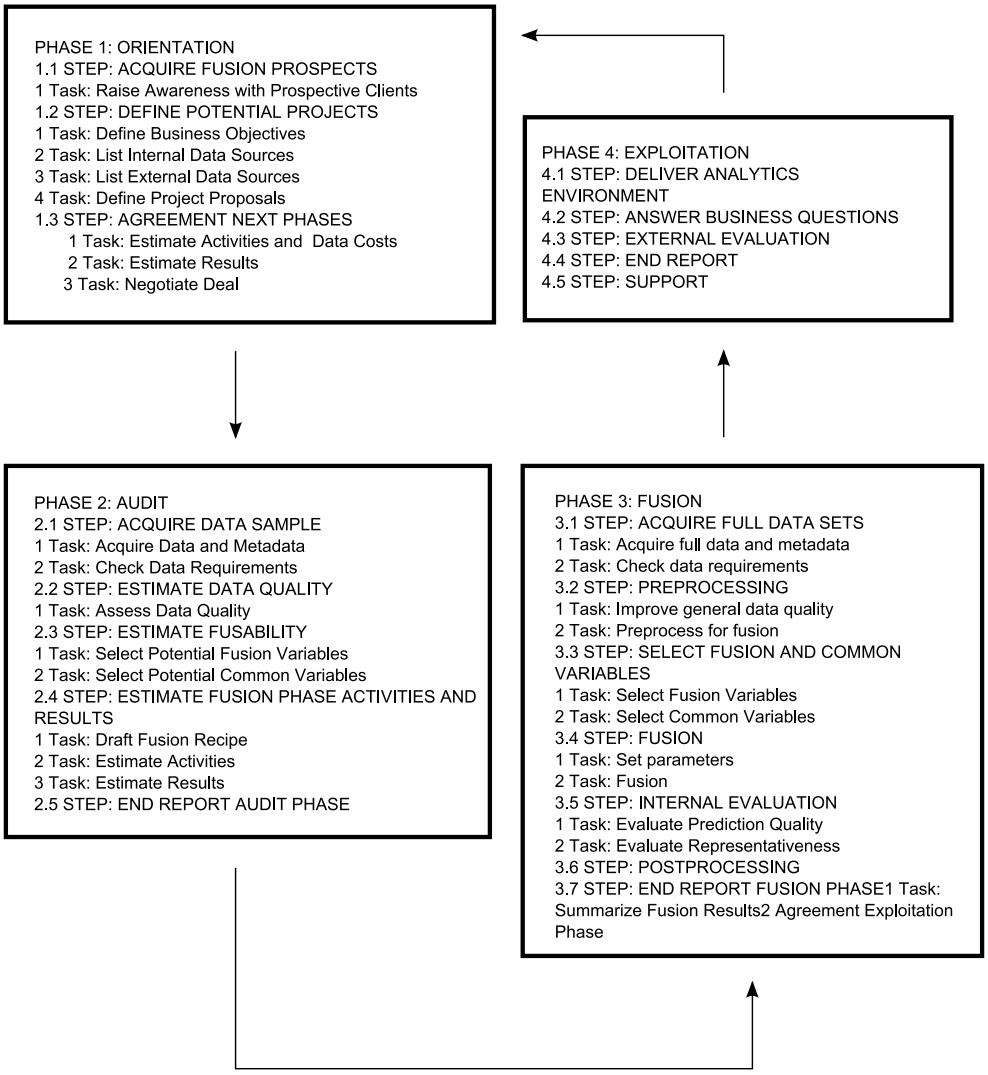


Figure 3.4: Fusion Factory Process Model

principle, the decision was made to develop a more detailed model of the end to end fusion process, for which we will provide summary highlights here.

There is no guarantee that fusion will always deliver added value. Data fusion projects are complex, with many steps and pitfalls. Instead of a single data set, several heterogeneous data sources are involved in the procedure that need to be mapped onto each other. Source data sets with hundreds to thousands of variables in a wide range of logical and physical formats are not uncommon. The fusion process itself consists of many intertwined phases and steps, and a lot of choices have to be made. What the right choices are is predominantly determined by factors outside the core fusion procedure, namely the business and data mining goals for which the enriched data set will be used.

Despite these challenges, we envision a streamlined fusion procedure where the core steps can be carried out in less than a working week instead of weeks or months (the current best practice in media research), using a predictable, reproducible process. To standardize and structure fusion projects we decided to develop a data fusion process model, borrowing some key concepts from data mining process models such as CRISP-DM (Chapman et al. 1999). The end goal of the fusion process model is to rationalize the process and automate it where possible, ultimately to the extent that end users of the fusion service could parameterize, control and execute large parts of it themselves. The development of the process model took place in parallel with three major data fusion projects carried out by a commercial data mining research company, Sentient Machine Research, for a financial services company, a charity and a marketing data provider. Further input was provided by previous experimentation on a variety of data sets and some 25 data fusion cases from research literature.

The high-level structure of the process model can be found in figure 3.4 and is described in detail in van der Putten, Ramaekers, den Uyl & Kok (2002). Four main phases have been identified, each of which will terminate in a go/no go decision. The first phase covers the scoping and definition of the project, including the data mining tasks for which the enriched data set will be used and a description of the donor and recipient data. Then an audit step takes place, in this phase the available data sets are analyzed separately and data quality and 'fusability' is assessed, through a variety of methods. On a go decision the actual fusion takes place including all internal evaluation activities. In the final phase, the enriched data set is being as an input to the regular data mining process to assess external quality. Ideally this then leads to further iterations of the overall process.

The process model could be used by an analyst to follow a structured approach towards carrying out fusion projects. It applies to database marketing but is generic enough to be extended to other domains. Alternatively, it can be used as a blueprint of the overall process to analyze where bottlenecks arise and to provide end to end process automation, or identify sub problems to be covered by data mining research. The model can easily be generalized to application areas other than marketing.

3.5 Conclusion

In this chapter we started by discussing how the information explosion provides barriers to the application of data mining and positioned data fusion as a possible solution to the data availability problem. We presented an overview of the main approaches adopted by researchers from outside the data mining and database marketing communities and described a database marketing case, for which a data set that was enriched by data fusion was used to predict propensity for credit card ownership. Our work is to our knowledge the first published case that discusses the value added by data fusion for predictive data mining, and also the first example to discuss data fusion in a database marketing context rather than market research or the fusion of surveys.

We hope to have shown that, despite its difficulties and pitfalls, the application of data fusion increases the value of data mining, because there is more integrated data to mine. Data mining algorithms can also be used to perform fusions, but publications on methods other than the standard statistical matching approach are relatively rare. Therefore we think that data fusion is an interesting research topic for knowledge discovery and data mining research.

From a database marketing and managerial point of view it will allow marketeers to bring information together from all kinds of sources in the organization, no matter how small the sample for which the information was gathered. This resulting data can be used for one to one marketing at individual customer level, rather than aggregate market research analysis, as if one could have extensive interviews with each of its millions of customers, at a fraction of the cost of real surveys. That said, there is no such thing as a free lunch, further research will be required to avoid overestimating the validity of the fused data and develop mining algorithms that appropriately deal with uncertain data.

Chapter 4

Bias-Variance Analysis of Real World Learning

Data mining research benchmark data sets are not always representative for real world problems. Similarly, studying the performance of modeling algorithms over data sets in 'laboratory conditions' may not suffice to explain differences in real world data mining results. In this chapter we present an analysis of the results of a large scale data mining competition we organized. It can be seen as a real world experiment to study data mining in the wild, based on noisy data, very limited time to deliver results, high competitive pressure and pragmatic incentives at stake rather than the usual requirement for following a proper scientific methodology.

There is a very large spread in the results for the prediction task in the competition and we use bias variance analysis as a framework to study the results and provide potential explanations for these differences. Bias variance analysis is usually only applied for characterizing modeling algorithms, but in this chapter we use it as a framework to evaluate the data mining process end to end (van der Putten & van Someren 1999), (van der Putten & van Someren 2000), (van der Putten & van Someren 2004).

4.1 Introduction

Data mining competitions have become increasingly popular. Competitions are organized for various reasons, such as unifying the research community and promoting the field to the outside world. In addition, competitions provide information on how data mining problems are solved in practice, when the goal is to solve the problem rather than to analyze the performance of a new method.

In one of these, the CoIL Challenge 2000, a prediction problem was used with properties that appear often in real world problems: noisy, skewed, correlated and high dimensional data with a weak relation between input and target. Extensive published results are available for the CoIL Challenge 2000. On request, 29 out of 43 participants provided a public report on how they solved the problem (van der Putten & van Someren 2000). We have also provided the results of 2 groups of in total 43 students (van der Putten & van Someren 2004). Reports for 6 entries for an earlier edition of this competition can be found in van der Putten & van Someren (1999). With the notable exception of the PKDD Discovery Challenge (Berka 1999), most competitions such as the KDD Cup only provide detailed reports on the top entries. The focus on best results only actually limits the study of data mining in the wild, especially for explaining practices that actually lead to worse results.

The objective of the competition was to predict who will be interested in a particular insurance product, a caravan policy, and to provide an explanation of why people would be interested. In this chapter we focus on the prediction task. Prediction models were used to select potential policy owners from a test set. The performance of the submitted solutions varied over a wide range, from one to two and a half times the number of policy owners that would have been found by a random selection and up to half of the maximum number of policy owners possible.

The main question we address here is what caused this wide range of performance. To explain the results we will evaluate the various approaches using bias-variance decomposition (Geman et al. 1992, Friedman 1997, Kohavi & Wolpert 1996, Breiman 1996, James 2003). This separates the error component resulting from the inability of a learner to represent or find the appropriate model for the behavior from the error component resulting from variance in predictions due to differences in models caused by sampling. Usually, bias-variance analysis is limited to the core modeling step, but we also apply it to other steps in the knowledge discovery process, such as attribute construction and selection.

This chapter is structured as follows. First the CoIL Challenge competition, problem tasks and data set are introduced (section 4.2). Then we present a general overview of the results for the prediction task (section 4.3). Section 4.4 provides more details on the method we used for analyzing the challenge problem and solutions, including bias-variance decomposition. Sections 4.5 and 4.6 focus on steps in the data mining process, data preparation and model development. Section 4.7 summarizes the expert evaluation of the description task of the competition. Finally we discuss the lessons learned (section 4.8).

4.2 Competition, Problem and Data Description

The CoIL Challenge 2000 was organized by the Computational Intelligence and Learning (CoIL) cluster, a cooperation between four EU funded research networks.

The goals of the challenge were to promote the application of computational intelligence and learning technology to real world problems, to clarify the relation between different approaches and to stimulate the search for solutions that combine different methods. The competition ran from March 17 to May 8, 2000 and was organized by the author and Maarten van Someren. Only just after the challenge deadline it was decided to publish the submitted solutions (van der Putten & van Someren 2000).

The objective of the competition was to predict who would be interested in buying a caravan insurance policy, and to give an explanation why people would buy. The problem was selected because it is representative of an important class of real world learning problems: domains with noisy, correlated, redundant and high dimensional data with a weak relation between input and target. Back then this kind of problem was not very well represented in benchmark collections such as the UCI Machine Learning Archive. The UCI data sets tended to be more cleaned up and geared towards illustrating the strengths of particular machine learning algorithms rather then being motivated by real world problems. The challenge data is now part of the KDD section of the UCI Archive (Blake & Merz 1998). The problem was split into a prediction and a description task.

4.2.1 Prediction Task

From a business perspective the goal of the prediction task is to rank current customers of the insurance company according to the probability that they will buy a caravan policy, so that the highest ranking customers can be contacted through a direct marketing campaign, for instance a mailing.

Only data about policy ownership is available, so it is assumed that owning this policy from the company is a reasonable approximation to buying the policy in response to a mailing. Given that only 6% actually owns the policy, regular zero-one loss or classification accuracy is not an appropriate evaluation metric. A model that predicts that no one will buy has a high classification accuracy of 94% but is useless for ranking and selecting customers. This illustrates that some default methods that are standard in machine learning research are not always directly applicable to real world prediction problems (see also section 2.1.5).

From a modeling perspective, the objective of the prediction task is to construct a model that assigns each customer of the insurance company a probability (or at least a probability rank score) that he will buy a caravan policy. If the costs of a mail piece and the profit of a mailing response are known the marketing analyst can then determine the optimal volume of customers to be mailed. However, in practice costs and benefits are not always known and in addition the behavior that is being modeled (ownership) differs from the actual behavior of interest (mail response). So we simplified the business case to the situation where there is a predetermined budget for a mail selection of 20%. The participants had to find the 800 clients in a

test set of 4000 instances who were most likely to have a caravan policy. The test set was given to the participants (without the target variable, caravan ownership). The performance metric was the number of correctly identified caravan policy holders among the 800 selected cases (or 20% of the total number of customers). The test set contained 238 policy holders.

Learning methods that construct models that only predict a class but not the probability may not be optimal for this problem, even if other loss functions than zero one loss are used. A classification model may classify less than 800 cases as a caravan policy owner. Adding random cases to fill up the selection will not give an optimal solution. Furthermore, if a model selects more than 800 cases, without a score or a probability there is no way to prioritize these cases to extract the best 800.

We could have opted for imposing an evaluation metric based on a given predefined loss matrix, or could have chosen an evaluation metric that evaluates the quality over the entire range of volume cut-offs, such as area under the ROC (AUC). However, the evaluation metric above was both simple to execute for participants and more representative for real world business metrics. We left it as an exercise to the participants to translate this business objective into the proper data mining approach and model evaluation metrics, to raise the bar and make the simulation more realistic.

4.2.2 Description Task

The purpose of the description task is to provide insight into why customers have a caravan insurance policy. This not necessarily the same as explaining the model underlying the predictions. Participants can use different approaches and algorithms for the description and prediction task. Descriptions can be based on prediction models but also on simple tables or parts of models. Given that the value of a description is inherently subjective, a domain expert from insurance marketing evaluated the submitted descriptions. The descriptions and accompanying interpretation were scored on comprehensibility, usefulness and actionability, for a marketing professional with no prior knowledge of machine learning.

4.2.3 Data Characterization

The effect of how steps in the analysis process are performed depends on properties of the data. As pointed out by Wolpert & MacReady (1995), heuristic methods or algorithms can only be optimal on a subset of all problems. In this section we provide a characterization of the problem and data.

The data that was available for both tasks consists of 5822 training instances and 4000 test instances[1]. The data contain 83 numeric and 2 symbolic input attributes

[1]The CoIL Challenge 2000 problem is also referred to as The Insurance Company (TIC) benchmark.

and the target, caravan policy yes/no, was only made available to the participants for the train set. The relation between input and target is very weak. The key attributes to explain policy ownership are not present in the data and measurement of input is noisy for reasons explained later in this section. The input can be divided in socio-demographics (43 attributes) and product ownership data (42 attributes). There are no missing values and all continuous attributes have been discretized into at most twelve ranges.

The socio-demographic information is linked to the postal code of the customer rather than to the individual customer. For instance a value of 5 for the attribute 'Home Owner' means that presumably 50-62% of people living in the same postal code area as this client own a house. The sociodemographics include information on marital status, household composition, education levels, employment types, social class and religion. Given that these attributes are linked to a single hidden variable, geography, these attributes may be highly correlated among each other. Measurement noise is also high. The marketing information provider collects the socio demographics by fusing information from various, possibly conflicting sources. Furthermore, it is certainly possible that customers living in the same area or in similar areas differ with respect to policy ownership. In spite of these obvious limitations, this kind of data is still very common for business to consumer marketing applications, especially if the responsible department can only access some very general internal information about its customers.

The product ownership data provide an overview of the product portfolio of the customer with the insurance company. For 21 policy products, the number of policies owned and amount of revenue is given. It is also very skewed: for 36 out of 42 attributes over 90% of the instances falls into the majority interval; only 6% actually owns a caravan policy. Typically, if the purpose is to predict customer behavior, this kind of behavioral data is a lot more predictive than traditional marketing segmentation variables based on internal company data such as age, gender etc., and even more predictive than external zip code demographics.

The TIC data differ from typical data sets in UCI ML archive. Table 4.1 shows that the TIC data set is relatively high dimensional. Furthermore we measured predictive power of input attributes by measuring information gain with respect to the class. The average predictive power of all attributes is low compared to UCI sets. The rightmost column of table 4.1 gives the average information gain of the five most predictive attributes. This shows that the average predictive power for the top five attributes is also low compared to UCI sets. The ratio between average predictive power for the top five attributes versus all attributes is relatively large, so only a small proportion of the attributes seem to matter. These differences will have consequences for the performance of different methods.

Data sets, problem descriptions, background info and reports can be found on the benchmark homepage at http://www.liacs.nl/ ~putten/library/cc2000

	instances	*input attributes*	*avg. info gain*	*avg. top 5 info gain*	*ratio top 5 vs avg*
TIC	5822	85	0.002	0.017	6.91
German credit	1000	20	0.017	0.045	2.62
hypothyroid	3772	29	0.024	0.132	5.45
breast cancer	286	9	0.034	0.056	1.66
pima	768	8	0.064	0.088	1.37
anneal	898	38	0.097	0.373	3.85
mushroom	8124	22	0.195	0.481	2.47
glass	214	9	0.369	0.511	1.39
soybean	683	35	0.455	0.974	2.14

Table 4.1: Some general features of the CoIL Challenge data (unbalanced train set) and selected UCI data sets

4.3 Overview of the Prediction Results

In this section we give an overview of the results for the prediction task. In total 147 participants registered, 43 sent in a solution to both tasks and 29 supplied a public report or permission to publish the supplied solutions. The sample of published reports is skewed: only 3 out of the top 50% best performing entries did not supply a report compared to 12 in the bottom half. Still this collection of reports provides a more accurate representation of successes and failures than regular research papers or most other competitions, for which generally only the best results are published. We encouraged participants to report on the entire solution path, including approaches that didn't seem to work. Entries came from both industry (31%) and academia (59%; remainder unknown) and included participants at various skill levels.

A wide variety of methods were used including instance selection, attribute selection, -construction and -transformation, hold out testing, cross validation, bootstrapping and ensemble learning, and cost sensitive classification; core prediction algorithms used included logistic regression, discriminant analysis, Naive Bayes, neural networks, support vector machines, evolutionary algorithms, genetic programming, fuzzy classifiers, RBF networks, self-organizing maps, decision trees, decision tables, rule based systems, ILP based methods and others.

The frequency distribution of scores for the prediction task is displayed in figure 4.1. To repeat, the participants had to select the 800 most probable caravan policy owners from a test set of 4000 instances (see section 4.2.1). The maximum number of policy owners that can be found is 238 (all owners in the test set), the winning model selected 121 policy owners. Random selection results in 48 policy owners (6% of 800).

The performance of the submissions varies over a wide range, from one (i.e.

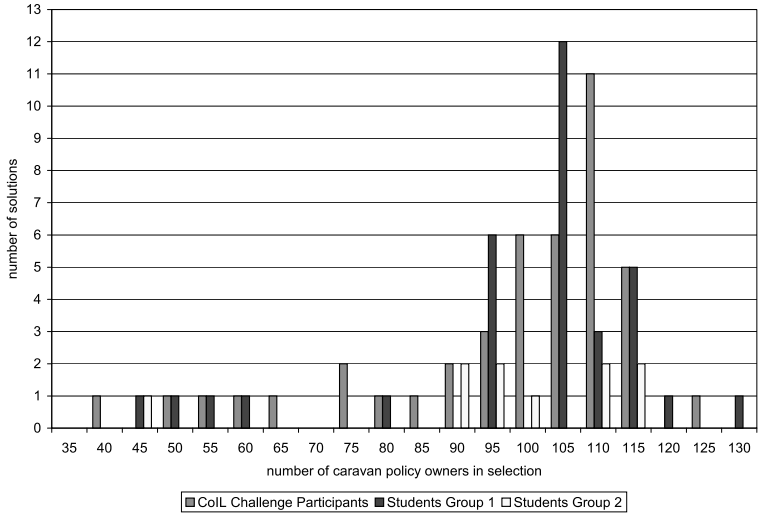


Figure 4.1: Histogram of prediction task performance for CoIL Challenge participants and two reference groups of students (bucket size is 5)

not better than random) to two and a half times the number of policy owners that would have been found by random selection and up to half of the maximum number of policy owners possible. These results may seem surprising, given the relatively small differences or improvements that are usually reported on UCI data for instance.

Figure 4.1 also displays the performance of two reference groups of students who worked on this problem after the competition as an assignment for a data mining course. The first group of students did not receive the test set targets nor were they informed of the CoiL Challenge or any of the results. In contrast, the second group of students read a paper written by the winner of the prediction task (Elkan 2001). Both groups compete very well with the CoIL participants on this problem. This an interesting result, given that these students were new to data mining. We will suggest some explanation for this in the discussion.

4.4 Meta Analysis Approach

The purpose of this study is to explain the results of the different solutions of the CoIL Competition to better understand the factors that determine the success of real world data mining projects. We organize the analysis by the main steps in the data mining process. According to the CRISP process model the top-level knowledge discovery process consists of business understanding, data understanding, data preparation,

modeling, evaluation and deployment (Chapman et al. 1999). Neither the business and data understanding step nor the evaluation and deployment steps were major parts of the prediction task of the competition. Therefore we focus on the data preparation (attribute construction and selection) and modeling steps.

For each solution to the competition task we collected the following data: prediction accuracy (number of caravan policy owners in test selection), if attribute construction was used (and if so, which method was used), attribute selection (and method), learning method and representation of the result, performance on the description task (comprehensibility, usefulness and actionability).

One characteristic property of this problem is the large noise component in the data. This means that overfitting is likely to be an important source of prediction errors. To analyze this effect we use the concept of bias-variance decomposition. When a model is constructed by a learning method from a sample taken from a given domain, and the model is used to make predictions then some predictions are false. Bias-variance decomposition distinguishes between (1) the *bias error*, a systematic component in the error associated with the learning method and the domain, (2) the *variance error*, a component associated with differences in models between samples and (3) an *intrinsic error* component associated with the inherent uncertainty in the domain. The intrinsic error is the variance within each point in the instance space, the error for the Bayes optimal classifier. High variance error indicates varying, unstable predictions and is associated with overfitting: if a method overfits the data the predictions for a single instance will vary between samples.

The concept of bias-variance decomposition was introduced to machine learning for mean squared error (Geman et al. 1992) and later versions for 'zero-one-loss' (predictions are correct or false) were given by Friedman (1997), Kohavi & Wolpert (1996), Breiman (1996) and Domingos (2000) and James (2003). Here we use the definition of Kohavi & Wolpert (1996). The expected misclassification rate (or expected classification cost where an error has cost 1 and a correct prediction cost 0) $E(C)$ is defined as:

$$E(C) = \sum_x P(x)(\sigma_x^2 + bias_x^2 + variance_x) \tag{4.1}$$

with

$$bias_x^2 \equiv \frac{1}{2} \sum_{y \in Y} [P(Y_F = y|x) - P(Y_H = y|x)]^2 \tag{4.2}$$

$$variance_x \equiv \frac{1}{2}(1 - \sum_{y \in Y} P(Y_H = y|x)^2) \tag{4.3}$$

$$\sigma_x^2 \equiv \frac{1}{2}(1 - \sum_{y \in Y} P(Y_F = y|x)^2) \tag{4.4}$$

where X is the instance space with elements x and Y the predicted variable, with elements $y \in \{0, 1\}$, the actual target function f a conditional probability distribution $P(Y_F = y_F|x)$, the hypothesis or model h generated by the learner a similar distribution $P(Y_H = y_H|x)$. In this model the bias error quantifies the difference between predicted and observed values over all predictions for a given x.

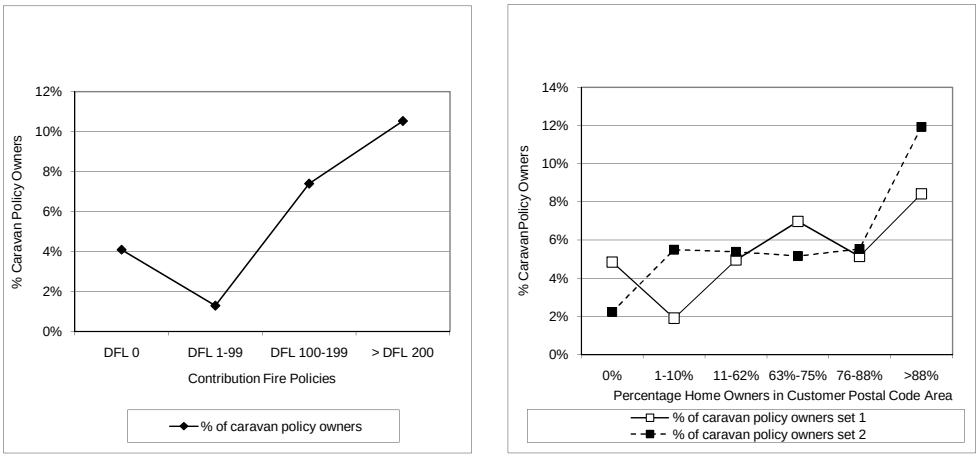
The intrinsic error and bias error can't be estimated separately. Here we mostly compare the bias and variance of different methods and in that case the intrinsic error contributes as a constant factor. The combined bias / intrinsic error effect and the variance error are estimated using the implementation in the WEKA toolkit (Witten & Frank 2000) following (Kohavi & Wolpert 1996). The data are split into two parts. From one part samples are drawn, learning is applied and the prediction error on the other half is calculated. We used 50 samples of size 200. The average error is the estimate for the *bias-inherent* error component and the variance between predictions estimates the variance component in the error.

To analyze the differences between methods we reconstructed a number of solutions to estimate the bias-variance decompositions. For these experiments we used data sets with balanced distributions because random sampling leaves a too few buyers instances and does not allow reliable estimates of the error components.

In general there is a trade-off between the strength of learning bias and overfitting. Methods with a strong learning bias are less likely to overfit, because their results depend less on the data samples used. However, if the learning bias of a method is not correct for a domain than this bias will be a source of prediction errors. Unlike the prediction errors caused by overfitting, errors caused by incorrect learning bias will not decrease with more data. In contrast, low bias learners are more flexible in terms of the ability of fitting complex patterns, but this comes at a risk of fitting noise, not signal and less evidence may be available to estimate parameters, both resulting in higher variance, which will only worsen if the amount of data available decreases. We can now characterize learning methods and also operations as attribute selection and attribute construction by the effect that they have on the bias and variance components in the error.

4.5 Lessons Learned: Data Preparation

The data step is a key step in the data mining process, and not only because typically a lot of time is spent in this step. We will see later in this section that it is also a key factor in the final quality of models. It is not uncommon that the impact of data preparation is larger than the impact of the choice of classifier used. Let us start with illustrating the role of bias and variance with some intuitive examples using the competition data. In figure 4.2(a) we have plotted the relationship between contribution (revenue) for fire policies versus caravan policy ownership. For sake of the argument let us assume that this is a correct estimate of the true relationship,



(a) (b)

Figure 4.2: Illustration of bias and variance issues using CoIL Challenge data: (a) non-monotone, non linear relationship between fire policy contribution and proportion of caravan policy owners (b) varying proportions of caravan policy owner estimates for different samples.

for instance it is based on very many representative instances. As can be seen the relationship is non linear and non monotone, so a classifier with a linear learning bias such as logistic regression is at a disadvantage. Another example can be seen in figure 4.2(b). Let us assume that we would have a simple model that for a given instance simply outputs the proportion of caravan policy owners associated with the home ownership bin the customer belongs to. You can see that in this example the estimate of policy ownership is unstable over different samples, i.e. this is a model with high variance – a problem that will only worsen if we introduce more bins.

Data preparation can alleviate these issues, or if applied improperly, make things worse. In the remainder of this section we will review methods for attribute construction, transformation and selection in more detail and attempt to explain the influence of these steps for the CoIL problem using the concepts of bias and variance.

4.5.1 Attribute Construction and Transformation

Attribute construction can reduce bias error by relaxing representational or search bias of a method. A risk of adding constructed attributes is that the variance component in the error can increase. However, attribute construction can also reduce the variance error, even without changing the representation bias. Below, we will illustrate the effect of attribute construction and discuss the effect on challenge solutions.

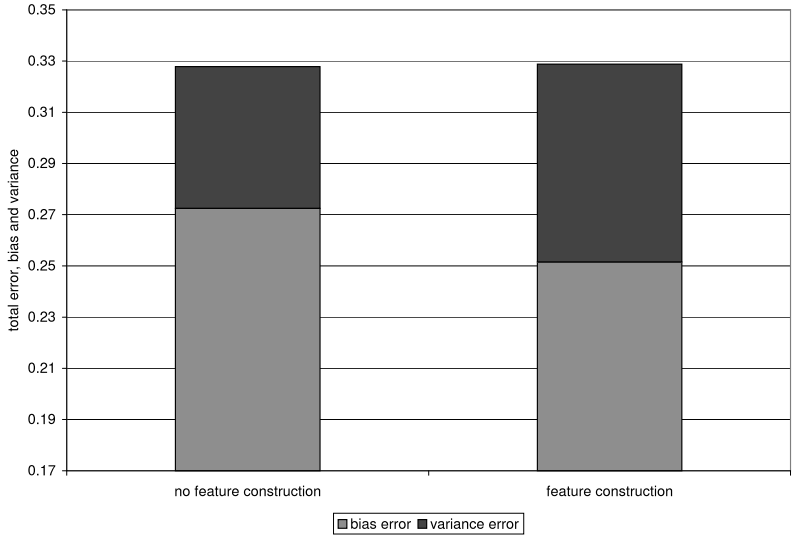


Figure 4.3: Bias and variance for winning model with and without constructed attribute

An example of attribute construction for avoiding representation bias is described in the winning entry by Charles Elkan (Elkan 2001). The algorithm he uses, naive Bayes, is limited in the sense that it can't model interactions between several input attributes. To compensate for this limitation he constructed a new attribute for each of the two most important products, car and fire policies, by taking the Cartesian product of the number of policies and the turnover amount. The new combinations replace the original attributes. Elkan claims that this was a vital contribution to his model. We repeated these experiments by comparing the bias and the variance of a Naive Bayes model for the original data and for the data with the constructed attributes. There is a clear reduction of bias, but balanced to a large extent by an increase in variance. So the reduction in total error is small.

Using the product of two attributes causes 'data fragmentation': it reduces the number of instances that are used for estimating the conditional probabilities in each cell, especially for values that already have low marginal frequencies. For a detailed discussion of conditions for the appropriateness of construction of Cartesian products for Naive Bayes and conditions for applicability of the bias of naive Bayes in general, see Domingos & Pazzani (1997).

Attribute construction can also reduce the variance component of the error. For instance, Jorgensen and Linneberg[2] first computed aggregate attributes such as the

[2]All public challenge submissions can be found in a tech report by van der Putten & van Someren (2000)

total number of policies and the total contribution. These were entered as attributes in Linear Discriminant Analysis and they were included in the discriminant function after pruning. This approach was found to have lower variance error than the standard method. In this case attribute construction does not relax representational bias but it lowers the variance error of the method, in particular the pruning step. Pruning decisions made about individual attributes are more sensitive to sampling variance than decisions about composite attributes, for example when one attribute represents the sum of several others. In this way, attribute construction will reduce the variance error. Less frequently used procedures for combined attribute construction and selection in the entries include principal component analysis and radial basis functions (see for instance the entry of Vesanto and Sinkkonen).

If we broaden the definition of attribute construction to include transformations on a single attribute the entry of White & Liu is also an interesting example. They cross-tabulated all predictors against the caravan policy ownership. All predictors showing 'substantial evidence of a quadratic relationship (or higher order polynomial) were recoded into constructed variables having a monotone relationship with the class' (van der Putten & van Someren 2000). This does not relax the representation bias of the learning method that they used, decision tree learning, but it does improve the search bias. This form of linearization reduces the number of intervals that must be constructed, making interval construction and pruning less sensitive to sampling and thereby this method reduces variance error. Various participants also use discretization to minimize the number of intervals. Apart from changing the attribute type from numeric to categorical, which may be needed because of practical limitations of the learner, these methods also aim at reducing the variance by decreasing the degrees of freedom.

We compared the five solutions with highest accuracy (over 110) with the five with the lowest accuracy (below 97). Of these only the best solution, by Elkan, constructed new attributes and used this to replace the original attributes. This suggests that for the TIC data, attribute construction is not critical. In our analysis of the winning entry, a reduction in bias is countered by an increase in variance (and vice versa). This risk should be taken into account when constructing or transforming attributes.

4.5.2 Attribute Selection

In theory, attribute selection can have various effects. Removing attributes that are irrelevant for the learner will not change the bias error. In other words, it will not change the loss in accuracy due to bias mismatch. As far as the learner is concerned, irrelevant attributes are noise. Attributes are irrelevant for a learner either because there is no real relation with the target, or the learning bias prevents capturing this relation. In the latter case intrinsic error may increase because information is lost. Attribute selection is generally aimed at variance reduction: fewer parameters need

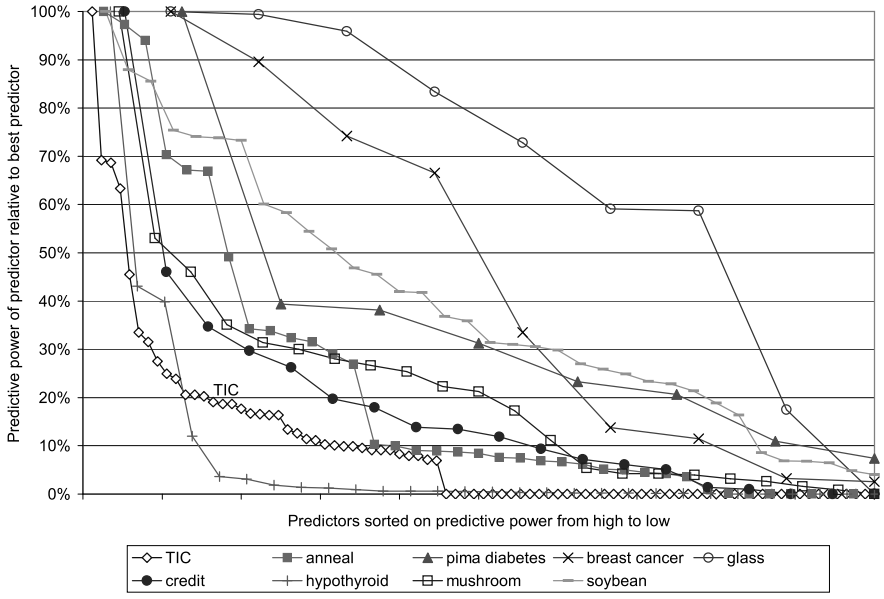


Figure 4.4: Distribution of predictive power for predictors, measured in information gain in proportion to the best predictor, for TIC (unbalanced training set) and a selection of UCI sets.

to be estimated, whereas the amount of relevant information that is removed is minimized. Removing relevant attributes may lead to an increase of intrinsic, bias and variance error.

The question is whether attribute selection plays a major role in the challenge prediction task. In section 4.2.3 and table 4.1 we have already shown that the TIC data set has very many attributes and that the predictive power of individual attributes is low measured in information gain (it is actually zero for more than half of the attributes). So the risk of overfitting the relation between individual attributes and the target is high and eliminating irrelevant attributes is likely to reduce the variance error.

Another indication is the predictive power of attributes relative to the best predictor. Figure 4.4 shows the information gain of attributes ordered from high to low, with the information gain scaled as percentage of the information gain of the most predictive attribute. For the TIC set, only a small proportion of the attributes has relatively high predictive power.

The effect of attribute selection on the final result will depend on the learning method that is used. If the learner itself selects attributes, a separate attribute selection step will have less effect. If we compare the submissions with highest and lowest accuracy, we see that four of the five highest scoring solutions used attribute selection

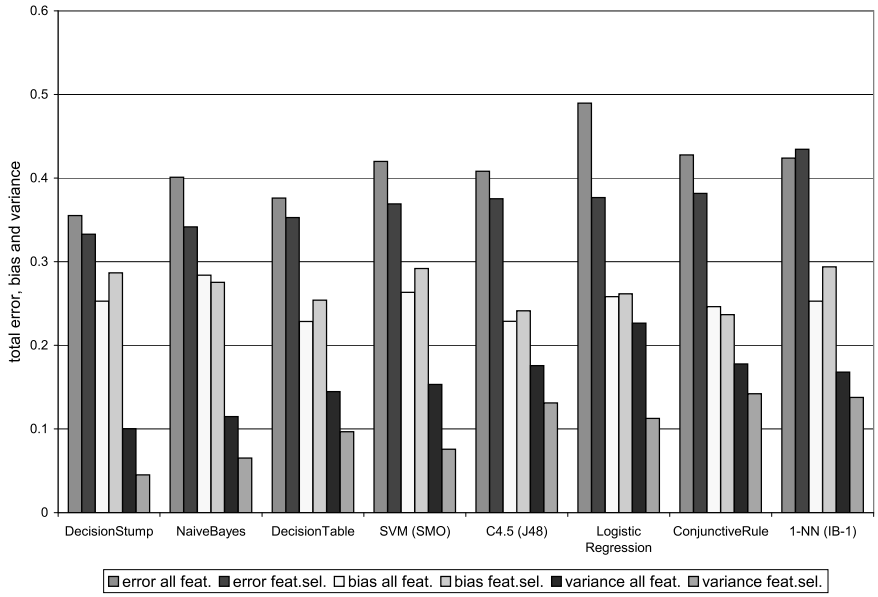


Figure 4.5: Bias-variance decomposition (bar graph) for eight learners with and without attribute selection.

and the fifth used a learner that eliminates irrelevant attributes. Of the five solutions with lowest performance only one includes separate attribute selection and two use a learning method that eliminates attributes.

To evaluate the relative importance of attribute selection, we performed bias-variance analysis on eight different learners, given the full set and a data set that was reduced to seven variables using the best first version of the CFS attribute subset selection algorithm (Hall 1999, Witten & Frank 2000). This algorithm takes both the predictive power of a predictor and its correlation to attributes that are already selected into account. To see the effects of extreme attribute selection, we included decision stumps (decision trees of depth 1, (Holte 1993)), which were not used by any participant. Note it generally produces a very limited range of scores which limits its practical use.

As can be seen from figure 4.5, attribute selection improves classification results for seven out of eight learners. When all 16 models are compared, six out of the top eight results are achieved using attribute selection. So for TIC using attribute selection or not seems to be more important than the choice of learning algorithm. Attribute selection reduces variance error for all eight learners.

The attribute selection methods used by the participants can be divided into three main categories (see Guyon & Elissee (2003) for an overview of the state of the art

in attribute selection). The first category consists of approaches for which candidate attributes are evaluated independently of other attributes. Simple evaluation measures such as correlation with the target attribute are used, and attribute selection is sometimes confirmed or guided by prior domain knowledge as well. For instance the author of the winning entry simply writes: 'As is common in commercial data mining, only data about the wealth and personal behavior of individuals is useful here'. He discarded all socio demographic attributes but one and the other five selected attributes were related to product ownership. The winner of the small scale Benelearn competition (van der Putten & van Someren 2000).that preceded the CoIL Challenge just used some simple cross tabulation to select the best attributes. The average performance of entries that were restricted to this kind of attribute selection (as far as known) was 99 caravan policies selected from the test set.

The second category contains methods that select subsets of attributes rather than just evaluating attributes individually and independently. There may be several reasons to look at subsets instead of single attributes (Kohavi & John 1997). An attribute with high individual predictive power but also high correlation to variables that are already selected does not add much information to the model, so it should not be included. An attribute with low individual predictive power may have some complex joint relationship with a selected variable that is highly predictive, in which case it may be advisable to include it. Several specialized subset attribute selection algorithms exist (e.g. Hall (1999)), however most participants in this category use a regular learner such as decision trees, decision tables or Naive Bayes to select attributes. We constrain this category to methods that use a different learner for attribute selection than for model development, so these are all so-called filter methods. The average performance in this category was 110 policy owners.

The third category are the so-called wrapper methods (see also Kohavi & John (1997)). These methods select subsets of variables using the same learner both for attribute selection and for model development. We use a broad definition for this category and include some participants that experimented extensively with manually selecting and deselecting attributes and retraining the learner. The average performance in this category was also 110 policy owners.

In conclusion, the majority of the participants use some form of attribute selection. The median number of attributes selected is 10 out of 85. Our experiments suggest that attribute selection is of key importance and that is likely to be a greater factor in determining the success than the choice of the learning method. The reason is that attribute selection is a powerful tool for variance reduction. Although the number of observations is small, attribute subset selection methods seem to outperform methods that evaluate candidate attributes individually, but for this problem we see no significant difference between subset filter and wrapper methods. This has been confirmed for other domains by a study on wrappers and filters (Tsamardinos & Aliferis 2003).

4.6 Lessons Learned: Learning Methods

The selection of a method for constructing the model of the data is generally considered an important decision in the data mining process, in addition to attribute selection and attribute construction. An important property of methods is the model representation. As discussed briefly in section 4.4 an inadequate learning bias of a method can cause bias error. An important characteristic of a method is therefore the strength and the content of the representation (or language) bias. Strong bias will in general reduce the error variance because it forces the learner into a small class of models. If the learning bias is incorrect then this will cause bias error.

The advantage of methods with stronger bias is typically lower variance. A way in which the representation affects the error variance is by its exploitation of redundancy. Models that involve (weighted) addition of attributes or of constructed predictors can exploit the fact that the noise in these predictors will average out and therefore the total prediction error will be smaller than that of the individual predictions. Also the proportion of training instances that is actually used to estimate model parameters is a factor: the more instances are used to estimate a parameter, the more stable an estimate will be. For instance nearest neighbor models and decision trees base a prediction on a small region in attribute space, in contrast to for instance logistic regression.

What is the effect of differences in learning bias between methods? The precise underlying pattern of the TIC domain is not known. The most predictive attributes are 'level of car insurance', 'number of car insurance policies', 'purchasing power class', 'level of fire insurance'. These attributes are correlated and their relation with the target class is approximately monotonic, although not completely, see section 4.5.1. This means that naive Bayes, rule-based, additive feature combinations and ensembles are all adequate representations. Because of the uncertainty in the relation, additive models are most attractive. Non-linear relations make naive Bayes and rule-based representations competitive. Because of the correlation between attributes, subsets of two or three of these (possibly discretized) variables give comparable optimal models within most model representations.

Table 4.2 shows the mean and maximum accuracies of the solutions of methods that are based on different model representations. Although we must interpret these data with care because of the small numbers, this suggests that differences in accuracy between model representations are relatively small. It is interesting to note that naive Bayes performs quite well although it has a relatively strong representational learning bias compared with the other methods. The best solution using naive Bayes included attribute construction and attribute selection and this may have corrected the representational learning bias.

Many authors mentioned that they experimented with a number of learning tools, and parameters of tools, but that the first results obtained sometimes turned out to

	Bias strength	*Additive*	*Bias correctness*	*Mean*	*Max*	n
Naive Bayes	strong	yes	good	118	121	2
Rule-based	weak	no	good	100,5	112	13
Additive linear	strong	yes	medium	111	111	2
Non-linear	weak	good	good	106	115	8
Ensembles	medium	yes	medium	112	115	1

Table 4.2: Representation bias and accuracy of CoIL solution methods

be the best. Experimentation can cause 'procedural bias' (Quinlan & Cameron-Jones 1995), (Domingos 1997): a new method is tried, or a variation of an earlier method and if the accuracy increases then it is assumed that the new method is better.

This may not be true because the new method may have a weaker bias or more degrees of freedom, allowing a closer fit to the data but with weaker support for the learned model. For example, consider first learning 'decision stumps', trees of depth one, and then decision trees of depth two. If we would simply compare the accuracies of these two methods, we would probably prefer the trees of depth two because these achieve higher accuracy but the support for the predictions of trees of depth two, for the path from root to leaf, is weaker for the deeper trees and the variance error and risk of overfitting will increase. Therefore only comparing the accuracies is not enough. Using cross validation can resolve this problem but only when the test set is very large. Cross validation itself relies on a sample and is therefore subject to error. As an aside we note that Domingos (1997) claim that overfitting is caused by testing too many hypotheses on a single data set is not the main problem. For example, trying out more candidate decision stumps will only improve the quality of the decision stump that is found. The problem is in the comparison of hypotheses (or classes of hypotheses) with different complexities or degrees of freedom. Usually a learning process that involves more hypothesis testing involves more comparisons between hypotheses with different degrees of complexity (or degrees of freedom) and therefore the risk of incorrect decisions is increased.

Also if more and more experiments are run with additional configurations, learners or parameter settings, but only the 'best' result is kept, the risk increases that we have found a spurious result. Just consider the informal definition of a significance level: the probability that a certain better result has happened purely by chance; in other words if we increase the number of configurations tested this is more likely to occur. Similarly, repeating runs with the same configuration over different samples contains this risk.

Because we do not have enough data about the number and nature of these

Accuracy	*Method*
121	Naive Bayes
115	Ensemble of 20 pruned naive Bayes
112	GA with numerical and Boolean operators
111	SVM regression
111	SVM regression
96	subgroup discovery
96	decision tree
80	fuzzy rules
74	CART
72	neural nets

Table 4.3: Accuracy and method selection

experiments, we focus on the methods that were used. Table 4.3 compares the five methods used by the participants with the highest scores with the five with the lowest scores (for which the method is known with enough detail). These results show that methods with weak learning bias tend to have low accuracies. A possibility is that participants have been seduced by the high accuracies that can be obtained by learners with weak learning bias and did not realize that the models found in this way can't be directly compared with models found by methods with stronger learning bias.

We explore the effect of model representation shape by reconstructing some of the solutions. Specifically, we can look at estimates of bias error and variance error in the CoIL solutions. The results of the bias-variance decomposition can be seen in bar chart and scatter plot format in respectively figure 4.5 and figure 4.6. The average bias is 0.25 ± 0.018 and 0.27 ± 0.023 without and with attribute selection, whereas the average variance is 0.16 ± 0.039 and 0.10 ± 0.036 without and with attribute selection. The bias component is relatively large because its estimate here includes the (constant) intrinsic error and it increases slightly through attribute selection. The variance component is smaller, but varies a lot more, and also decreases a lot more through attribute selection. The rule-based methods have a variance component in the error of around 0.20. The most stable results are produced by naive Bayes which has a variance error of only 0.09.

Concluding, the results show that for the TIC domain the bias error is more stable between methods than the variance error. This suggests that selection or even improvements of methods should be found in reducing the variance component, rather than the bias component.

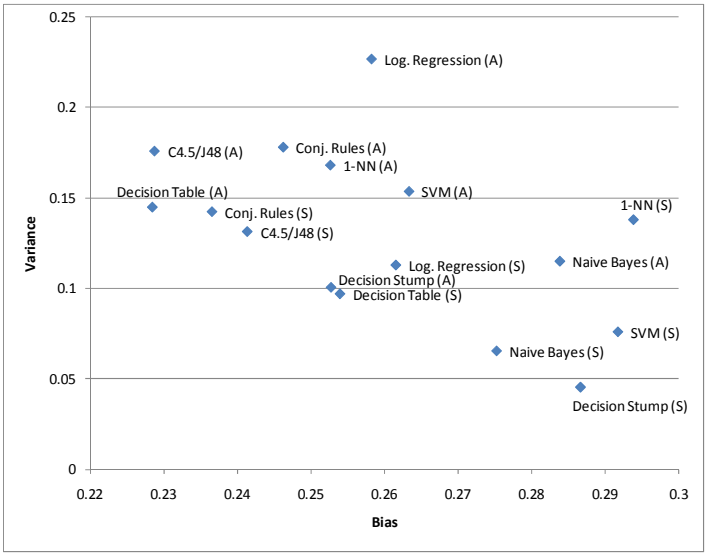


Figure 4.6: Bias-variance decomposition for eight learners (scatter plot) with (S) and without attribute selection (A).

4.7 Lessons Learned: Description Task

The goal in the description task was to explain why people own a caravan policy, given the data, modeling methods and subjective, domain-based interpretation. The descriptions and accompanying interpretation had to be comprehensible, useful and actionable for a marketing professional with no prior knowledge of computational learning technology.

Submitted descriptions were evaluated by a marketing expert (Stephan van Heusden from MSP Associates in Amsterdam). The expert commented: "Almost all entries lack a good description in words: participants seem to forget that most marketeers find it difficult to read statistics (and understand it)". The expert stressed the importance of actionability. "A good entry combines a description of the results with a tool to use the results." Participants from industry had a better score than academic participants (4.3 versus 3.5 out of a maximum of 6), although these differences were not significant given the standard deviations (1.6 resp. 2.1). Similar to the prediction solutions, a wide variety of approaches was chosen, although there was a tendency to use simple statistics, cross tabulations and rule based solutions.

The winners, Kim and Street, used a variety of techniques for the description task, including proprietary evolutionary algorithms, chi square tests and association rules. The marketing expert remarked: "This participant clearly explained which steps

preceded his conclusions. He may have discovered all conclusions the others also found, but additionally he really tries to interpret them." The expert also appreciated the association rule results, the discussion of the complex nonlinear relation between purchasing power and policy ownership and the explanations why people were not interested in a caravan policy. Their prediction model scored 107 policy owners (65th percentile).

The success of the winning entry demonstrates that to explain behavior the prediction models used do not necessarily need to be of top quality. This was confirmed by analyzing the correlation between prediction and description scores for all entries. Rather than giving a complete overview of a single model, good results are achieved by applying multiple methods and choosing the most comprehensible, useful and actionable patterns from these models.

4.8 Discussion and Conclusion

The CoIL competition resulted in analyses of a real-world problem by a number of experts. We used the bias-variance decomposition of errors to identify causes of success and failure in solving the competition problems. The TIC problem is characterized by a relatively large number of intercorrelated attributes, a lot of uncertainty and skewed distributions of inputs and target. This is common for many real world problems. In this case variance error was a larger problem than bias error.

Attempts to discover complex models using methods with weak learning bias and weak methods to avoid overfitting lead to complex models that were unstable and that overfit. The best approach is to simplify the data set through data preparation first and then use simple, robust, strong bias methods for modeling. This suggests a potential reason why the CoIL experts didn't outperform the students. Apparently a simple model and experimental setup suffice to solve this noisy prediction problem.

4.8.1 Lessons

To summarize we would like to single out the following lessons learned for problems similar to the challenge:

1. In the case of noisy prediction problems choices in the analysis process should be aimed at reducing the variance component rather than at finding an appropriate bias.

2. The potential impact of data preparation (e.g. selecting the right attributes) on variance and overall error reduction is larger than the choice of classifier for these high variance problems.

3. Attempting to improve the fit between the bias of the problem and method by using complex learners is only useful when parameters can be estimated reliably. For problems such as the one of the CoIL Challenge 2000, simple models with few degrees of freedom are most robust so these should be tried first.

4. All steps of the data mining process risk increasing variance error. This suggests that model stability should be tested: if a method, applied to different samples, produces different models with different predictions, this suggests that variance error may be a problem. The method may be overfitting the data or may be otherwise unstable.

5. Measures against overfitting all have their weaknesses. Cross validation, being an empirical method, is not guaranteed to result in the best model, especially if it is used to limit the complexity of a model such as a decision tree. If intrinsic error is high and the amount of data is low, estimates of validation set performance are uncertain. Extensive experimentation while only keeping the 'best' models is also risky. Quite a few participants reported that they were seduced to spoil their original models (cf. Seewald; Abonyi and Roubos; Sathiya Keerthi and Jin Ong; Kaymak and Setnes). They increased the fit on the data by additional heuristics or fine-tuning, ending up with a model that is worse rather than better than the original.

4.8.2 Further research

We identify a number of directions for further research:

1. Bias-variance decomposition is an elegant framework for the analysis of learning problems because it provides a diagnosis of error into various components. Each component requires a different strategy of error reduction, so the data miner has more insight into what action to take to improve the model. To become a standard tool, a more universally accepted definition of bias and variance is needed, both for zero one loss and for other evaluation metrics, such as asymmetric cost functions and the area under the ROC curve.

2. This study confirms the importance of steps before and after the core modeling step such as attribute construction, attribute selection and model evaluation. Bias-variance decomposition could be used more to analyze and improve methods used in these steps. Evaluation of bias and variance components should be integrated more tightly in methodology, methods and systems for Machine Learning. In addition to using the concepts of bias and variance for analysis, methods could be developed that explicitly minimize bias and variance error.

Chapter 5

Profiling Novel Algorithms

The introduction of new technologies generally follows a typical adoption (and hype) cycle, and novel data mining algorithms are no exception to this rule. Neural networks are a good example. Algorithms inspired by neural processing have been around since the forties (for instance McCulloch & Pitts (1943)) but really gained traction in the eighties of the last century after publication of the PDP Handbooks (Rumelhart & McClelland 1986). The amount of neural network research exploded and neural networks were pitched as a superior set of algorithms for classification, clustering and optimization, in some cases with no more justification than its biological origins.

After this period of excitement but also over-inflated claims a more realistic approach was taken. Some researchers went the direction of using neural networks strictly for the purpose of neurological modeling, but for non-biological modeling applications the data mining and machine learning community started rightfully to ignore the biological roots and evaluate and benchmark neural algorithms against other approaches using generally applicable measures such as accuracy. Whilst this may have resulted in the loss of some of the initial appeal, research interest and promise, it actually led to the incorporation of neural networks into the standard toolkit of a much wider community.

So for the maturity and wider acceptance of a novel algorithm it is key that it is benchmarked against and compared with existing approaches, preferably by researchers who have not been involved in the development and evangelization of the particular novel algorithm. However, it should be noted that basic accuracy benchmarking only provides worst case reassurance that the algorithm provides reasonably valid results and should not be used to make general claims about superiority of the novel method over others. The No Free Lunch theorem loosely states that there is no algorithm that will consistently outperform all other algorithms on all

problem domains (Wolpert & MacReady 1995). So it is important to take the analysis further than basic accuracy benchmarking, and for instance investigate on what kind of problems and data it works well, or to explore to what other algorithms it is similar in its behavior. This will help the data miner to decide when best to apply these methods. We refer to this as algorithm profiling rather than basic benchmarking. Existing methods such as learning curve analysis and bias variance analysis can be used, but there is also a lot of opportunity to develop new methods (see figure 5.1).

In this chapter we provide an example approach for such an analysis. As the candidate novel algorithm we have chosen for AIRS, a so called Artificial Immune System or Immunocomputing algorithm, a prediction method inspired by the learning capabilities of the immune system (Watkins, Timmis & Boggess 2004). The analogy with neural networks is not a coincidence; we wanted to pick a field that is likely to be in a similar position as neural networks previously. Whilst our approach goes further than basic benchmarking, we have chosen to keep it fairly straightforward and simple, so that the same approach can easily be used to benchmark, profile and characterize other novel algorithms for classification, and it will hopefully inspire researchers to develop new model profiling methods (van der Putten & Meng 2005), (Meng et al. 2005), (van der Putten et al. 2008).

5.1 Introduction

There has been a rapid growth in the interest in Artificial Immune Systems for applications in data mining and computational intelligence recently. The immune system is sometimes called the second brain for its abilities to recognize new intruders and remember past occurrences, and artificial immune systems lend concepts and mechanisms from natural immune systems for a variety of data mining and optimization applications (Castro & Timmis 2002), (Watkins et al. 2004).

Simulating the immune system or translating immune system mechanisms into machine learning is an interesting topic on its own. However, as discussed in the context of neural networks, to be accepted as a candidate algorithm for data mining applications rather than biological modeling, the source of inspiration for these algorithms is not really a topic of interest. Instead empirical evidence is needed that these algorithms produce high quality, reliable results over a wide variety of problems compared to a range of other approaches. Also any expert fine-tuning should be avoided. When benchmarking a novel algorithm there is a risk that the algorithm developer (even unintentionally) performs more and better tuning on the novel classifier as compared to the benchmark classifiers, as he has more experience with his own algorithm. Also, the purpose of a benchmark should be to compare results under normal, i.e. non expert, user conditions.

Given that we are interested in applicability of artificial immune systems for real world data mining, and that classification is one of the most important mining

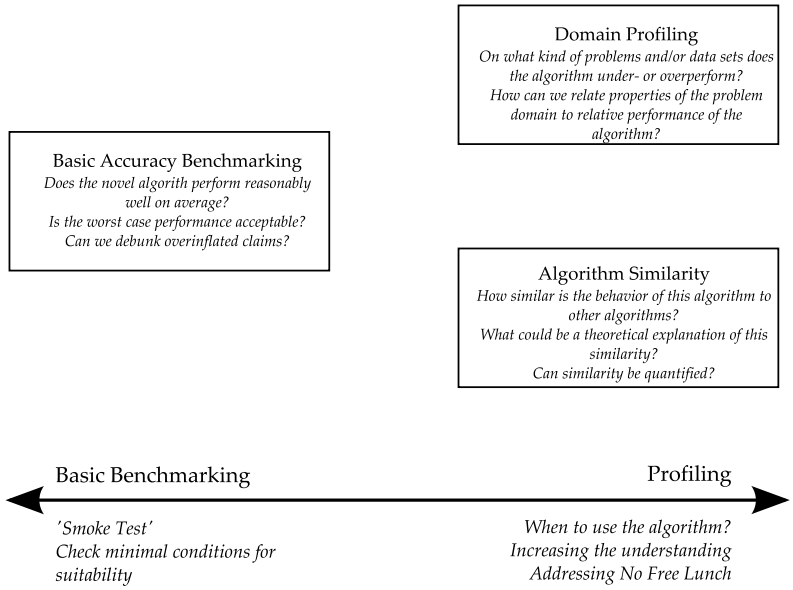


Figure 5.1: Basic algorithm benchmarking versus algorithm profiling

tasks, we focus on the Artificial Immune Recognition System (AIRS) algorithm. AIRS was introduced in 2001 as one of the first immune systems approaches to classification (Watkins 2001) and seemed to perform reasonably well on various classification problems. Until then, several papers have been published dealing with AIRS benchmarking (Goodman, Boggess & Watkins 2002), (Goodman, Boggess & Watkins 2003), (Marwah & Boggess 2002), (Watkins et al. 2004). However, in our opinion these approaches were relatively limited, given that comparisons were made on a small number of data sets and algorithms, and that the benchmark results were sourced from literature rather than produced under exactly the same conditions as for the AIRS algorithm.

In contrast to the previous work mentioned, all our experiments have been run from scratch, to guarantee consistent experimental conditions. This includes applying AIRS on a wide range of representative real-world data sets with large differences in number of instances, attributes and classes, and comparing its performance to a wide range of commonly accepted algorithms. This will provide an answer to the question whether AIRS is already mature enough to be considered as a generally applicable data mining algorithm – or whether indeed the performance of AIRS is even superior to other algorithms – as the existing AIRS research literature claimed at the time of publication of our basic benchmarking results (van der Putten & Meng 2005),

(Meng et al. 2005).

The accuracy benchmark can be seen as a 'smoke test' to check minimal conditions for suitability. For example, does the algorithm provide reasonably valid and robust results, so no critical outliers? As already argued, for the novel algorithm to be accepted as a valuable addition to the data miners toolbox, answering this question will not be sufficient. It should also be made clear in what situations the algorithm will likely be most applicable: for instance by identifying on what kind of data it will work well, or to what other algorithm it is similar, not just in theory but particularly in term of its behavior on real data (see figure 5.1).

We refer to this as algorithm profiling rather than basic benchmarking. In our opinion algorithm profiling is a useful umbrella term for a family of analyses that should be applied when new algorithms are introduced. Existing methods such as learning curves and bias variance analysis can be used, but there is also a lot of opportunity to develop new methods.

In the AIRS case, first we investigate the relationship between data set properties and algorithm performance, to get a better picture when AIRS may perform better or worse than others. We focus on the size of the data set, this can easily extended further to include other data set properties. We then investigate what other algorithms have a similar empirical behavior as the AIRS algorithm. As discussed, the aim of both these analyses is to provide a deeper understanding in what cases AIRS may be a valid algorithm to use.

The remainder of this chapter is organized as follows. Section 5.2 provides an overview of natural and artificial immune systems, and section 5.3 outlines the AIRS algorithm. The basic benchmarking, data set properties and algorithm similarity experiment results are described in sections 5.4, 5.5 and 5.6 respectively. We conclude the chapter with section 5.7.

5.2 Immune Systems

The recognition and learning capabilities of the natural immune system have been an inspiration for researchers developing algorithms for a wide range of applications. This section introduces some basic immune system concepts and provides the history and background behind the AIRS algorithm for classification.

5.2.1 Natural Immune Systems

The natural immune system offers two lines of defense, the innate and adaptive immune system. The innate immune system consists of cells that can neutralize a predefined set of attackers, or antigens, without requiring previous exposure. The antigen can be an intruder or part of cells or molecules of the organism itself. In

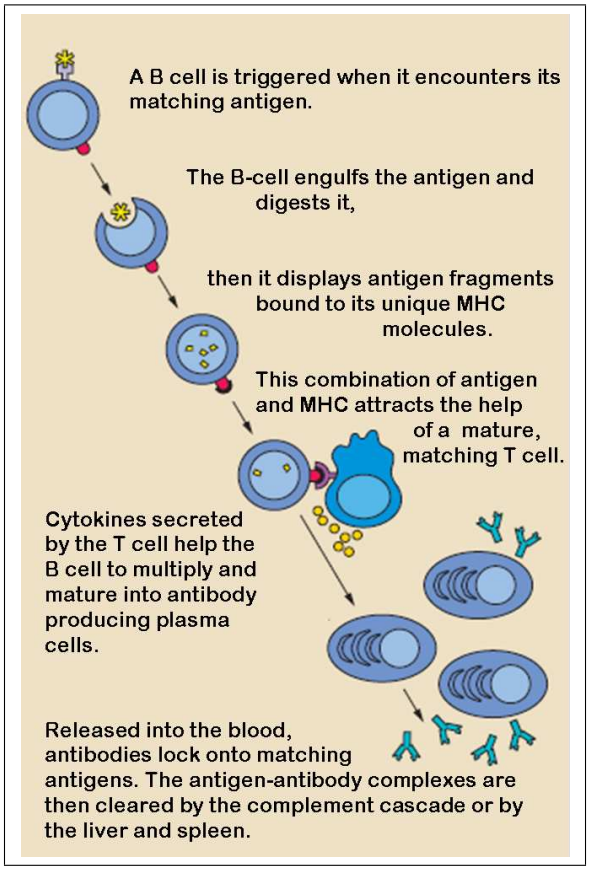


Figure 5.2: Primary immune response example (B-Cells). When a B cell encounters its triggering antigen, it gives rise to many large cells known as plasma cells, which essentially are factories for producing antibodies. Source Wikipedia "Adaptive Immune Systems", March 27 2008; NIH Publication No. 035423, September 2003 (modifications: September 4, 2006)

addition, higher animals such as vertebrates possess an adaptive immune system that can learn to recognize, eliminate and remember specific new antigens.

An important role is played by lymphocytes, cells that recognize and destroy antigens. There are different types of lymphocytes, cells that recognize antigens directly (B-cells) or cells that recognize antigens that are bound to so called presenter cells (T-cells). Each lymphocyte codes for a specific antigen, but there may be more possible types of antigens than there are specific lymphocytes.

This is solved by a form of natural selection. The bone marrow and thymus continuously produce lymphocytes and each of these cells can counteract a specific type of antigen. Now if for example a B-cell lymphocyte encounters an antigen it codes for, it will produce antibody molecules that neutralize the antigen and in addition a large number of cloned B-cells are produced that code for the same antigen (clonal expansion or clonal selection; see figure 5.2).

The immediate reaction of the innate and adaptive immune system cells is called the primary immune response. The immune system also keeps a record of past intrusions. A selection of the activated lymphocytes is turned into sleeper memory cells that can be activated again if a new intrusion occurs of the same antigen, resulting in a quicker response. This is called the secondary immune response (Castro & Timmis 2002).

5.2.2 Artificial Immune Systems and AIRS

Natural immune systems have inspired researchers to develop algorithms that exhibit adaptivity, associative memory, self - non self discrimination and other aspects of immune systems. These artificial immune system algorithms have been applied to a wide range of problems such as biological modeling, computer network security & virus detection, robot navigation, job shop scheduling, clustering and classification (Castro & Timmis 2002).

The Artificial Immune System algorithm (AIRS) can be applied to classification problems, which is a very common real world data mining task. Most other artificial immune system research concerns unsupervised learning and clustering. The only other attempt to use immune systems for supervised learning is the work of Carter (Carter 2000). The AIRS design refers to many natural immune system metaphors including resource competition, clonal selection, affinity maturation, memory cell retention, and so on. AIRS builds on the concept of resource limited clustering as introduced by Castro & von Zuben (2000) and Timmis & Neal (2001).

According to the introductory paper, AIRS seems to perform well on various classification and machine learning problems (Watkins 2001). Watkins claimed the performance of AIRS is comparable, and in some cases superior, to the performance of other highly-regarded supervised learning techniques for these benchmarks.

Later on, Goodman, Boggess, and Watkins investigated the ‘source of power

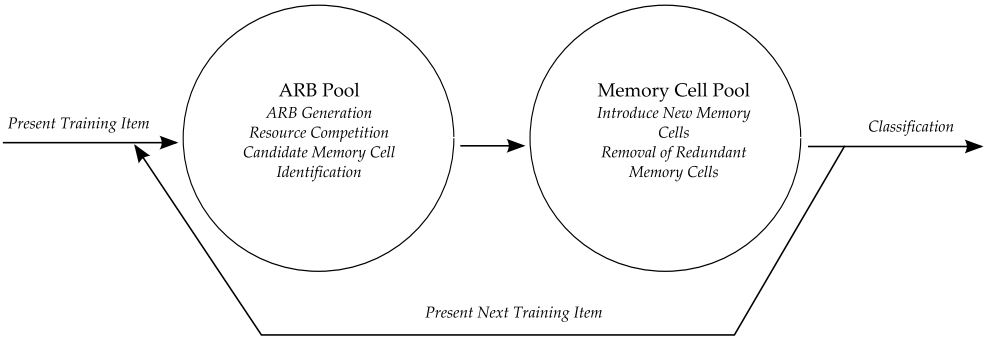


Figure 5.3: An overview of the AIRS algorithm

for AIRS' and its performance on multiple-class problems. The authors compared the results of AIRS on several data sets including iris, ionosphere, diabetes, sonar, and Cleveland heart with the results from a large number of other classifiers taken from literature. Based on this comparison, the authors claim "AIRS is competitive with the top five to eight classifiers out of 10-30 best classifiers on those problems", "it was surprisingly successful as a general purpose classifier" and it "performed consistently strong across large scope of classification problems" (Goodman et al. 2002), (Goodman et al. 2003). In Marwah & Boggess (2002), the authors investigated several technical problems for AIRS and found that on the e-coli data set AIRS produced a higher average accuracy than any other published result. In Watkins et al. (2004), the authors investigated the modifications in the mechanisms of memory cell evolution and somatic hypermutation and concluded that these improved both the performance and simplicity in comparison to results from other algorithms in literature.

5.3 AIRS: the Algorithm

From a pure data mining point of view, AIRS is a cluster-based approach to classification. It first learns the structure of the input space by mapping a codebook of cluster centers to it and then performs a k-nearest neighbor search on the cluster centers for classification, just like k-means clustering for classification or Self Organizing Maps (SOMs, (Kohonen 1982)). The attractive point of AIRS is its supervised procedure for discovering both the optimal number and position of the cluster centers, which it shares with some more rarely applied approaches for competitive learning such as NeuralGaz (Martinetz 1993), (Martinetz & Schulten 1994) and Fritzke's Growing Self Organizing Neural Networks (Fritzke 1994), (Fritzke 1995).

In AIRS, there are two different populations, the Artificial Recognition Balls (ARBs – lymphocytes) and the memory cells, see figure 5.3. When a training antigen is presented, ARBs matching the antigen are activated and awarded more resources. ARBs with too few resources will be removed and new ARBs are created through mutation. This corresponds to the primary immune response in natural immune systems. On convergence a candidate memory cell is selected which is inserted to the memory cell pool if it contributes enough information. This corresponds to the secondary immune response. This process is repeated for all training instances – each training item can be seen as a separate 'attack'. Classification takes place by performing a nearest neighbor search on the memory cell population (Watkins 2001), (Watkins et al. 2004). Below we will describe AIRS in more technical detail.

5.3.1 Initialization

Let us assume we have a training data set X containing n labeled instances $ag_i = \{x_i, t_i\}$ with x_i an input with d attributes and t_i a one dimensional target class (i=1,2,. . . ,n). First all the data items will be normalized so that the affinity of every two training instances ag_i and ag_j is in the range [0,1]. Second the average affinity between all training instances is calculated. The average affinity is called the affinity threshold:

$$\text{affinity threshold} = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} \text{affinity}(x_i, x_j)}{\frac{n(n-1)}{2}} \tag{5.1}$$

with x_i and x_j the attribute vectors of the ith and jth training antigens, and affinity(x, y) returns the Euclidean distance between the two antigens attribute vectors.

We assume the set MC to be the memory cell pool containing m memory cells: $MC = \{mc_1, mc_2 \ldots, mc_m\}$, and set AB as the ARB-population containing r ARBs: $AB = \{ab_1, ab_2, \ldots, ab_r\}$, with $mc_j = \{x_j^{mc}, t_j^{mc}\}$, $j = (1, 2, \ldots, m)$; $ab_k = \{x_k^{ab}, t_k^{ab}\}$, $k = (1, 2, \ldots, r)$. On initialization, the memory cells pool MC and the ARB population AB are seeded by randomly adding training instances.

5.3.2 Memory Cell Identification and ARB Generation

From now on, antigens (training instances) will be presented to the algorithm one by one. If an antigen $ag_i = \{x_i, t_i\}$ is presented to the system, the algorithm will identify a memory cell $mc_{match} = \{x_{match}^{mc}, t_{match}^{mc}\}$ which has the same class label ($t_{match}^{mc} = t_i$) and is most stimulated by the specific ag_i. The stimulation function is simply calculated as:

$$S(ag_i, mc_j) = 1 - \text{affinity}(x_j, x_j^{mc}) \tag{5.2}$$

with affinity is defined as Euclidean distance.

If there is no mc_{match} available at this moment, just let ag_i act as the mc_{match}. This mc_{match} will then be cloned to produce new m_c clones. First the attributes of mc_{match} will be mutated with a certain probability (default 0.1). If any mutations occurred for this particular clone, the class label will be mutated as well with the same probability

5.3.3 Resource Constrained ARB Evolution

At this moment, there are a set of ARBs including mc_{match}, mutations from mc_{match}, and others from previous training. AIRS mutates these memory cell clones to generate new ARBs. The number of ARBs allowed to produce is calculated by the product of the hyper clonal rate, clonal rate (both default 10), and the stimulation level S. The newly generated ARBs will be combined with the existing ARBs.

AIRS then employs a mechanism of survival of the fittest individuals within the ARB population. First, each ARB will be examined with respect to its stimulation level when presented to the antigen. In AIRS, cells with high stimulation responses that are of the same class as the antigen and cells with low stimulation response that are not of the same class as the antigen are rewarded most and allocated with more resources. The losers in competing for resources will be removed from the system. Then the ARB population consists of only those ARBs that are most stimulated and are capable in competing for resources.

Then the stop criterion is evaluated. The stop criterion is reached if the average stimulation value of every class subset of AB is not less than the stimulation threshold (default 0.8). Then the candidate memory cell $mc_{candidate}$ is chosen which is the most stimulated ARB of the same class as the training antigen ag_i. Regardless whether the stop criterion was met the algorithm proceeds by allowing the ARBs the opportunity to proliferate with more mutated offspring. This mutation process is similar to the mutation of phase 2, with a small exception: the amount of offspring than to be produced is calculated by the product of stimulation level and the clonal rate only. If the evaluation criterion was not met in the last test, the process will start again with the stimulation activation and resource allocation step. Otherwise the algorithm will stop.

5.3.4 Memory Cell Pool Update

Now if $mc_{candidate}$ is more stimulated by the antigen than mc_{match}, it will be added into the memory cell pool. In addition, if the affinity value between $mc_{candidate}$ and mc_{match} is also less than the product of the affinity threshold (average affinity between all training items) and the affinity threshold scalar (a parameter used to provide a cut-off value, default 0.8), which means $mc_{candidate}$ is very similar to mc_{match}, $mc_{candidate}$ will replace mc_{match} in the set of memory cells. Training is completed now for this training instance ag_i. and the process is repeated for the next instance.

5.3.5 Classification

With the training completed, the evolved memory cell population $MC = \{mc_1, \ldots, mc_m\}$ ($m \leq n$) will be used for classification using k-nearest neighbor. The classification for a test instance will be determined by the majority vote of the k most stimulated memory cells.

5.4 Basic Accuracy Benchmarking

The goal of the benchmark experiments is to evaluate the predictive performance of AIRS in a real world application setting. We assume that our users are non data mining experts, e.g., business users, who may lack knowledge or time to fine-tune models, so we used default parameters wherever possible to create a level playing field. For this reason we also decided to use simple accuracy rather than more advanced measures such as area under the ROC. To ensure consistency, the experiments for all classifiers were carried under exactly the same conditions, in contrast to some earlier published work on AIRS (see section 5.2.2).

5.4.1 Approach

We selected data sets with varying number of attributes, instances and classes, from simple toy data sets to difficult real world learning problems, from the UCI Machine Learning and KDD repositories (Blake & Merz 1998). 'Breast cancer' refers to the Wisconsin Breast Cancer variant of the UCI breast cancer data set. The TIC data sets are derived from the standard TIC training set by downsampling the negative outcomes to get an even distribution of the target. In addition, TIC5050S only contains the most relevant attributes according to a subset attribute selection method (van der Putten & van Someren 2000), (van der Putten & van Someren 2004), (Hall & Holmes 2003), (Hall 1999).

In the experiments, we selected some representative, well known classifiers as challengers. These classifiers include naive Bayes, logistic regression, decision tables, decision trees (C45/J48), conjunctive rules, bagged decision trees, multi layer perceptrons (MLP), 1-nearest neighbor (1-NN) and 7-nearest neighbor (7-NN). This set of algorithms was chosen because they cover most of the algorithms used in business data mining and correspond to a variety of classifier types and representations – instance based learning, clustering, regression type learning, trees and rules, and so on. Furthermore we added classifiers that provide lower bound benchmark figures: majority class simply predicts the majority class and decision stumps are decision trees with one split only. For AIRS we chose the 1 and 7 nearest neighbor versions of the algorithm. We used the Java version of AIRS by Janna Hamaker (Hamaker

& Watkins 2003) and the WEKA toolbox for the benchmark algorithms (Witten & Frank 2000).

All experiments are carried out using 10-fold stratified cross validation. The data is divided randomly into ten parts, in each of which the target class is represented in approximately the same proportions as in the full data set. Each part is held out in turn and the classifier is trained on the remaining nine-tenths; then the classification accuracy rate is calculated on the holdout validation set. Finally, the ten classification accuracy rates on the validation sets are averaged to yield an overall accuracy with standard deviation. To test the robustness of classifiers under real world conditions, all classifiers were run with default settings, without any manual fine-tuning.

5.4.2 Results

The results of the experiments can be found in table 5.1. With respect to the worst case classifiers we highlight some interesting patterns. Almost all classifiers outperform majority vote. The comparison with decision stumps (single split decision trees) as a worst case benchmark is more striking. For example, for all data sets with the exception of the waveform data set the conjunctive rules classifier does not perform better than decision stumps. Other examples are the TIC data sets: none of the classifiers other than C45 and Decision Tables on TIC5050S perform better than decision stumps. This demonstrates the power of a very simple decision rule in a real world black box modeling environment (see also Holte (1993)).

To get a better picture of the relative performance of AIRS we compare it to the average classifier performance (excluding decision stump and majority vote). AIRS-1 performs better than average on 3 of these 9 datasets. AIRS-7 performs better than average on 6 of these 9 datasets. This conflicts with the claims made in earlier studies that were cited in section 5.2.2 on superior performance of AIRS.

We also made some comparisons to the IB-k algorithms, because these may be closest to a trained AIRS classifier. AIRS-1 improves on IB-1 more often than the other way around; this is probably due to the fact that AIRS-1 provides some useful generalization. However IB-7 performs better than AIRS-7 on all of the data sets. AIRS-7 performs better than AIRS-1 on 7 out of 9 data sets. Using more clusters may give better results but not to the extent that IB-7 can be beaten (basically as many cluster centers as data points).

That said, with the exception of AIRS-1 on German credit data, the AIRS algorithms produce at least around average results. Given this benchmark, the previous claims about supposedly exceptional performance of AIRS seem to be, perhaps not surprising, somewhat overinflated. However our results do suggest that AIRS is a mature classifier that delivers reasonable, robust performance and that it can safely be used for real world classifications tasks, which is to our opinion a very positive, if not the most positive result that could be expected for a novel algorithm.

	Sonar	**Wisc. Breast Cancer**	**Wave form**	**Iris**	**Iono-sphere**	**Pima dia-betes**	**Ger-man credit**	**TIC 5050**	**TIC 5050s**
Maj. Class	*53.4* ± 1.7	*65.5* ± 0.5	*33.8* ± 0.1	*33.3* ± 0.0	*64.1* ± 1.4	*65.1* ± 0.4	*70.0* ± 0.0	*49.7* ± 0.4	*49.7* ± 0.4
1NN	**86.6** ±7.0	95.3 ±3.4	73.6 ± 1.3	95.3 ± 5.5	86.3 ± 4.6	70.2 ± 4.7	72.0 ± 3.1	*55.9* ± 7.8	59.9 ± 5.1
7NN	80.8 ± 7.8	**96.6** ± 2.2	80.1 ± 1.1	**96.7** ± 3.5	85.2 ± 4.3	74.7 ± 5.0	74.0 ± 4.1	61.1 ± 3.2	65.4 ± 8.4
Dec. Stump	73.1 ± 8.3	*92.4* ± 4.4	*56.8* ± 1.5	*66.7* ± 0.0	*82.6* ± 4.8	71.9 ± 5.1	*70.0* ± 0.0	**68.5** ± 4.7	**68.5** ± 4.7
C45 (J48)	71.2 ± 7.1	94.6 ± 3.6	75.1 ± 1.3	**96.0** ± 5.6	**91.5** ± 3.3	73.8 ± 5.7	70.5 ± 3.6	**68.1** ± 5.5	**69.1** ± 4.4
Naive Bayes	*67.9* ± 9.3	96.0 ± 1.6	80.0 ± 2.0	**96.0** ± 4.7	*82.6* ± 5.5	**76.3** ± 5.5	**75.4** ± 4.3	62.8 ± 6.4	68.0 ± 3.3
Conj. Rules	*65.9* ± 8.7	*91.7* ± 4.5	*57.3* ± 1.3	*66.7* ± 0	*81.5* ± 5.4	*68.8* ± 8.67	*70.0* ± 0	**67.4** ± 3.7	68.3 ± 4.5
Bag-ging	77.4 ± 0.1	95.6 ± 3.1	**81.8** ± 1.4	94.0 ± 5.8	**90.9** ± 4.4	74.6 ± 3.6	**74.4** ± 4.9	59.9 ± 5.8	68.4 ± 4.1
Log. Regr.	73.1 ± 13.4	**96.6** ± 2.2	**86.6** ± 2.3	**96.0** ± 5.6	88.9 ± 4.9	**77.2** ± 4.6	**75.2** ± 3.4	62.7 ± 4.6	66.5 ± 3.4
MLP	**82.3** ± 10.7	95.3 ± 2.6	**83.6** ± 1.7	97.3 ± 3.4	**91.2** ± 2.8	**75.4** ± 4.7	71.6 ± 3.0	60.7 ± 4.3	65.4 ± 4.7
Dec. Table	74.5 ± 8.2	95.4 ± 2.7	73.8 ± 1.6	92.7 ± 5.8	89.5 ± 4.5	73.3 ± 3.6	72.2 ± 4.1	61.9 ± 4.5	**69.1** ± 5.7
AIRS1	**84.1** ± 7.4	96.1 ± 1.8	75.2 ± 1.7	**96.0** ± 5.6	86.9 ± 3.1	*67.4* ± 4.6	*68.0* ± 5.1	*56.8* ± 4.4	*55.0* ± 6.5
AIRS7	76.5 ± 8.4	**96.2** ± 1.9	79.6 ± 2 .2	95.3 ± 5.5	88.6 ± 5.0	73.6 ± 3.5	71.4 ± 3.1	57.8 ± 5.5	*59.1* ± 6.1

Table 5.1: Average accuracy and standard deviation on accuracy (tenfold) for AIRS and a range of benchmark algorithms. Best results in boldface, worst results in italics.

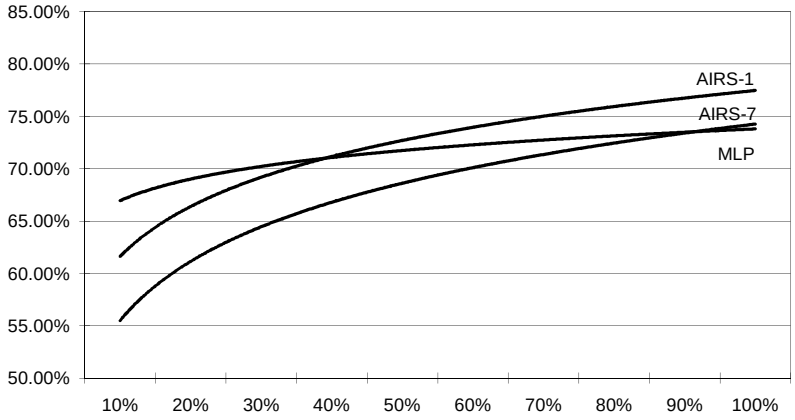


Figure 5.4: Accuracy learning curves for AIRS-1 and AIRS-7 and MLP

5.5 Profiling: Influence of Data Set Properties

As mentioned the No Free Lunch theorem states that there is no classifier that outperforms all other classifiers across all problem domains (Wolpert & MacReady 1995). So it is interesting to investigate on what kind of data AIRS performs relatively well and on what kind of data it won't, for example by relating data set properties to the performance of AIRS relative to other algorithms. We focus on a key property, the size of the data set.

5.5.1 Approach

We carried out a so called learning curve analysis on the diabetes data set. We created models using the same set of classifiers as in the previous section, for simplicity using a fixed 25% hold out test set. These experiments were carried out on training samples of varying size, starting with 10% and with 10% increments. Note that the results on the full training set can be different from the overall benchmark results given the differences in train and test set size and the simple hold out testing approach rather than full cross validation; for this test we are primarily interested in high level learning curve patterns. The learning curves were not all smooth monotonically increasing, to be able to spot trends we fitted logarithmic trend lines to the result series for each of the classifiers.

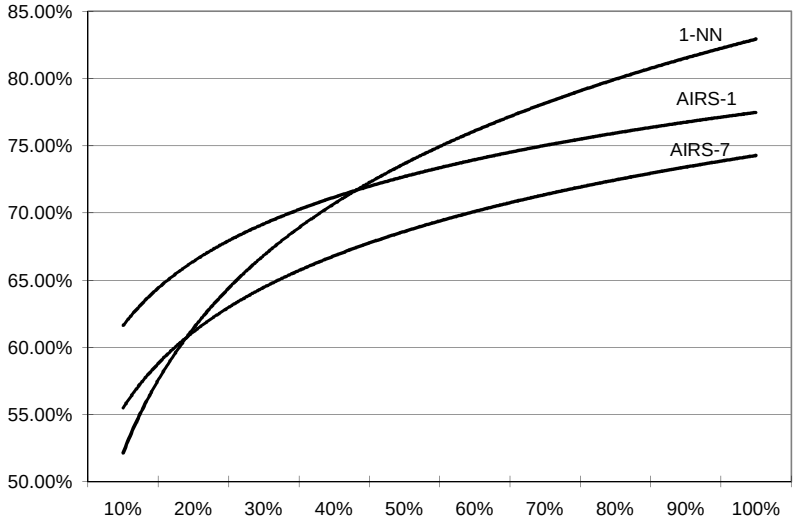


Figure 5.5: Accuracy learning curves for AIRS-1 and AIRS-7 and 1-NN

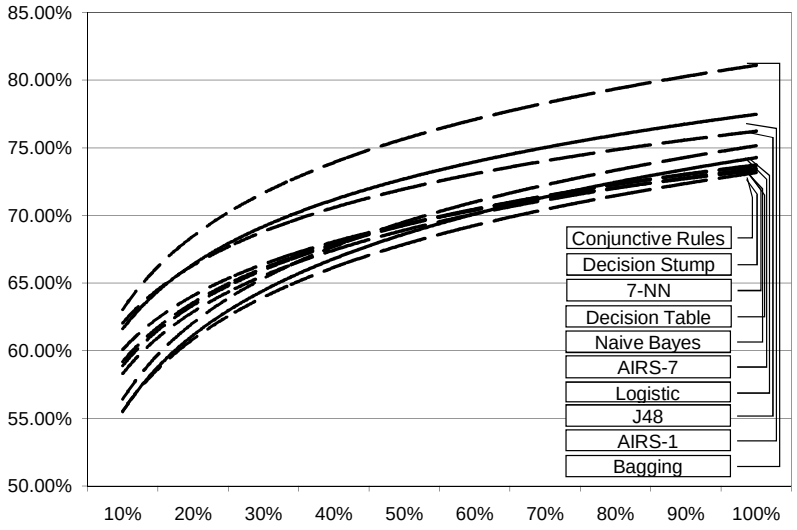


Figure 5.6: Accuracy learning curves for AIRS-1 and AIRS-7 and remaining classifiers

5.5.2 Results

Roughly three patterns of result series emerge. In figure 5.4 you can see the trend lines of AIRS-1 and AIRS-7 compared to MLP. The trend of AIRS-1 and AIRS-7 is quite similar: in this case AIRS-1 outperforms AIRS-7. Ignoring trend lines, only by looking at the data point for the full data set AIRS-7 actually performs better than AIRS-1, which is consistent with with the benchmark findings reported earlier. The trend of the MLP curve is a lot flatter, i.e. it outperforms AIRS at lower data set size but AIRS starts to perform better at larger data set sizes. The opposite is true for 1-NN. The slope of the 1-NN learning curve is steeper than the AIRS learning curves, see figure 5.5. For the remaining classifiers, including 7-NN, the learning curve has a similar curve as the AIRS classifiers (figure 5.6).

5.6 Profiling: Computing Algorithm Similarity

The experiments above provide an overview what the performance of the AIRS algorithm is and how AIRS performance may relate to data set size. Another typical question for a novel algorithm is how similar it is in its behavior compared to other algorithms. This is useful to know, as it will give a data miner an idea when to apply this technique.

Some insight can be derived by studying the theoretical properties of the algorithm. For instance, the AIRS learning process can be seen as a relatively advanced (but also complex) way to produce a simple codebook of labeled cluster centers, so in theory its behavior could be similar to nearest neighbor and k-means clustering style of classifiers.

The key question though is whether this behavior can be confirmed through experiments, and whether other classifiers of very different origins may behave similarly as well, for reasons yet to be determined.

5.6.1 Approach

In our experiments we have used three different ways to measure algorithm similarity. We focused on the accuracy of the algorithm, given that it is generally the key behavior of interest.

The benchmark provided the raw data for the analysis. To get a basic picture we simply calculated the correlation between classifiers on series of accuracies over the various data sets. A problem though with this approach is that to a large extent correlation can already be expected; on difficult problems accuracy will be low and vice versa.

So in our second method we decided to focus on performance relative to other classifiers. For each classifier – data set combination we evaluated whether perfor-

Classifier	Correlation AIRS-1
1-NN	0.99
7-NN	0.98
MLP	0.97
AIRS-7	0.97
Bagging	0.94
Decision Table	0.92
Logistic	0.89
J48	0.87
Naive Bayes	0.83
Decision Stump	0.47
Conjunctive Rules	0.46
Majority Vote	-0.07

Table 5.2: Correlation with AIRS-1 accuracy series

Classifier	Correlation AIRS-7
MLP	0.99
7-NN	0.99
Bagging	0.98
Logistic	0.97
AIRS-1	0.97
Decision Table	0.96
1-NN	0.95
Naive Bayes	0.93
J48	0.92
Conjunctive Rules	0.48
Decision Stump	0.45
Majority Vote	-0.04

Table 5.3: Correlation with AIRS-7 accuracy series

mance was better or worse than the average of all classifiers on that particular data set. We then counted how often classifiers agreed in terms of over or under performance with the AIRS algorithms. We have excluded the majority vote and decision stump classifiers from the calculation of the average given that these acted as worst case performance classifiers.

A drawback of this particular approach is that we lose how much better or worse a classifier was than average, in relative terms. So in our third approach we calculated the number of standard deviations a classifier under or over performed. To calculate similarity we then computed the correlations between these series of standard deviations.

5.6.2 Results

The correlation between the AIRS-1 accuracy series and the other algorithms can be seen in table 5.2. As mentioned in section 5.3.5 the AIRS algorithm can be seen as a codebook learning procedure that automatically determines the optimal number of codes. Classification is done by simple nearest neighbor search on the codebook. As expected the nearest neighbor classifiers indeed have a high correlation, along with the AIRS-7 algorithm, and the 1-NN algorithm indeed behaves more similar than the 7-NN algorithm. A somewhat unexpected result is the high score for MLP. The AIRS-7 results (table 5.3) show a consistent yet slightly more mixed picture with the 1-NN and AIRS-1 algorithms scoring lower and MLP ranking as the first algorithm. This could have been due to the lower variance of AIRS-7 (bagging scores high, AIRS-1 scores higher than 1-NN).

The results of our second method can be found in tables 5.4, 5.5 and 5.6. Note that in this method we only look at whether a method scores better or worse then

	1-NN	7-NN	J48	Decision Table	Naive Bayes	Conj. Rules
sonar	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE
breast cancer	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE
waveform	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
iris	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE
ionosphere	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE
diabetes	FALSE	TRUE	TRUE	TRUE	TRUE	FALSE
German credit	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE
TICTRAIN5050	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE
TICTRAIN5050s	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE
Times > average	2	7	5	5	7	2

	Bagging	Logistic	MLP	AIRS-1	AIRS-7
sonar	TRUE	FALSE	TRUE	TRUE	TRUE
breast cancer	TRUE	TRUE	FALSE	TRUE	TRUE
waveform	TRUE	TRUE	TRUE	FALSE	TRUE
iris	TRUE	TRUE	TRUE	TRUE	TRUE
ionosphere	TRUE	TRUE	TRUE	FALSE	TRUE
diabetes	TRUE	TRUE	TRUE	FALSE	TRUE
German credit	TRUE	TRUE	FALSE	FALSE	FALSE
TICTRAIN5050	FALSE	TRUE	FALSE	FALSE	FALSE
TICTRAIN5050s	TRUE	TRUE	TRUE	FALSE	FALSE
Times > average	8	8	6	3	6

Table 5.4: Relative performance of each classifier by data set (better than average for all classifiers; two tables above)

	Agreement AIRS-1
1-NN	8
AIRS-7	6
7-NN	5
Conjunctive Rules	4
Bagging	4
MLP	4
J48	3
Decision Table	3
Naive Bayes	3
Logistic	2

Table 5.5: Frequency of agreement (under or over performance compared to average) between AIRS-1 and other classifiers

	Agreement AIRS-7
Bagging	7
MLP	7
7-NN	6
AIRS-1	6
1-NN	5
Logistic	5
J48	4
Decision Table	4
Naive Bayes	4
Conjunctive Rules	0

Table 5.6: Frequency of agreement (under or over performance compared to average) between AIRS-7 and other classifiers

average, and we count how often classifiers agree. For AIRS-1 we do see the expected behavior with respect to the AIRS-7 and the nearest neighbor algorithms, MLP now scores lower, perhaps because the magnitude of error is now lost. For AIRS-7 we see a similar pattern, however nearest neighbor and AIRS are even less similar than bagging and MLP.

The results for the third method can be found in tables 5.7 (number of standard deviations difference from average accuracy) and 5.8 (respective correlations). For this method the most consistent pattern as per the expectations emerge with high scores for AIRS, nearest neighbor and MLP. This method also seems to gives the widest range in similarity scores which makes it easier to discriminate across classifiers (obviously assuming the score itself is valid).

In table 5.10 we provide an overview of the results for the various methods. Overall it can be concluded that classifiers with theoretical similarities (AIRS-*k*, *k*-NN) indeed also behave similar. In addition the MLP algorithm behaves similar, an interesting result that was unexpected.

5.7 Conclusion

In this chapter we have presented an approach to benchmarking and profiling a novel algorithm, in this case the AIRS artificial immune system algorithm. We are interested in artificial immune systems because it is one of the newest directions in biologically inspired machine learning, and as such subject to a keen interest from the community but also at the risk of being overhyped. We focused on AIRS because it can be used for classification, which is one of the most common data mining tasks.

This was the first basic benchmarking evaluation of AIRS that compared AIRS across a wide variety of data sets and algorithms, using a completely consistent experimental set up rather than referring to benchmark results from literature. In contrast to earlier claims, we find no evidence that AIRS consistently outperforms other algorithms. However, AIRS provides stable, around average results so it can safely be added to the data miners toolbox.

In addition we have presented a methodology for further profiling of a novel algorithm. We have performed some explorative learning curve experiments that showed a more or less standard curve for AIRS, steeper than MLP but flatter than nearest neighbor. We have explored a variety of methods for computing algorithm similarity that confirmed it behaved indeed similar to nearest neighbor based methods, but also similar to MLP.

Whilst sometimes even proper basic benchmarking is lacking for novel algorithms, as was also the case with AIRS, we propose that more attention is being paid to the area of algorithm profiling. Each novel method should in principle be subjected to both basic benchmarking as well as model profiling. Given its importance and the fact that that there are not a lot of generally accepted standard methods for algorithm

	Majority Vote	1-NN	7-NN	J48	Decision Table	Naive Bayes	Decision Stump
sonar	-3.45	1.53	0.66	-0.78	-0.28	-1.28	-0.5
breast cancer	-21.86	-0.08	0.87	-0.6	0.03	0.44	-2.17
waveform	-5.57	-0.43	0.41	-0.24	-0.41	0.39	-2.61
iris	-6.77	0.28	0.43	0.35	-0.03	0.35	-2.98
ionosphere	-6.92	-0.36	-0.7	1.16	0.57	-1.45	-1.45
diabetes	-2.59	-0.97	0.49	0.2	0.03	0.99	-0.43
German credit	-0.97	-0.11	0.76	-0.75	-0.02	1.36	-0.97
TICTRAIN5050	-2.98	-1.4	-0.08	1.73	0.15	0.37	1.84
TICTRAIN5050s	-3.18	-1.05	0.1	0.88	0.87	0.64	0.76

	Conj, Rules	Bagging	Logistic	MLP	AIRS-1	AIRS-7
sonar	-1.57	0.15	-0.5	0.89	1.17	0.02
breast cancer	-2.7	0.13	0.87	-0.08	0.55	0.58
waveform	-2.54	0.62	1.24	0.85	-0.23	0.34
iris	-2.98	0.12	0.35	0.5	0.35	0.28
ionosphere	-1.79	0.99	0.4	1.07	-0.19	0.31
diabetes	-1.43	0.45	1.28	0.7	-1.85	0.11
German credit	-0.97	0.93	1.27	-0.28	-1.83	-0.36
TICTRAIN5050	1.54	-0.37	0.33	-0.18	-1.18	-0.92
TICTRAIN5050s	0.7	0.73	0.34	0.1	-2.07	-1.23

Table 5.7: Number of standard deviations difference between classifier performance and average, by data set (two tables above)

Classifier	Correlation with AIRS-1
1-NN	0.77
AIRS-7	0.66
MLP	0.43
7-NN	0.16
Bagging	-0.34
J48	-0.35
Decision Table	-0.48
Logistic	-0.50
Majority Vote	-0.52
Decision Stump	-0.54
Conjunctive Rules	-0.63
Naive Bayes	-0.66

Table 5.8: Correlation for relative under or over performance between AIRS-1 and other classifiers

Classifier	Correlation with AIRS-7
AIRS-1	0.66
MLP	0.52
1-NN	0.48
7-NN	0.23
Logistic	0.22
Bagging	0.07
Naive Bayes	-0.28
J48	-0.49
Decision Table	-0.55
Majority Vote	-0.56
DecisionStump	-0.85
Conjunctive Rules	-0.93

Table 5.9: Correlation for relative under or over performance between AIRS-7 and other classifiers

	Similar to AIRS-1	Similar to AIRS-7
Method 1	1-NN, 7-NN, MLP, AIRS-7	MLP, 7-NN, Bagging, Logistic
Method 2	1-NN, AIRS-7, 7-NN	MLP, Bagging, 7-NN, AIRS-1
Method 3	1-NN, AIRS-7, MLP, 7-NN	AIRS-1, MLP, 1-NN, 7-NN
Summary	1-NN, AIRS-7, 7-NN	MLP, 7-NN, AIRS-1

Table 5.10: Algorithm similarity for the computation methods

profiling yet, we think that the development of such methods can be an interesting independent area for research. The approaches offered in this chapter are just an exemplary starting point, given that the main purpose was to use a case example to demonstrate that model profiling can be an interesting, relevant and attractive area for data mining research.

Chapter 6

Summary and Conclusion

In this final chapter we provide a summary of conclusions, lessons learned and a vision for future research, given the overall theme of data mining in context. Below we will outline and relate some of the main findings of the chapters, see the chapters itself for more background and references.

The main purpose of the cases chapter (chapter 2) is to demonstrate that successful data mining applications involve a lot more than just applying or improving a core modeling algorithm. The first two cases were originally written for audiences with no data mining or computer science background, marketeers and medical professionals respectively. In both fields there is a clear push towards a more data driven, quantitative or even scientific approach, as illustrated by trends such as evidence based medicine, personalized treatments, database marketing, one to one marketing and real time decisioning. These cases provide an inside out, end to end view of data mining and the data mining process, taking the application context rather than the technology as a starting point. One of the findings of both studies was that there were often no major differences between prediction algorithms on a problem, nor were there clear winners across the overall range of problems.

The third case is an even stronger example of this. This case is a more research oriented project dealing with the recognition of pathogen yeast cells in images. For this case it is still an open question whether the underlying problem is truly easy to solve (classifying yeasts) given that the data mining problem is trivial (classifying pictures resulting from the experimental set up chosen). This is a good practical example that the particular translation of the research or business problem into a data mining problem has a major impact on the results, which is a topic that should be covered by the first step in the data mining process, defining the objectives and experimental approach.

The medical case on predicting head and neck cancer survival rate points out

another specific issue. Whilst building a model on a single data set is relatively straightforward and may lead to models with comparable accuracy, a variety of other data sets from other sources is also available. These data sets may differ in terms of attribute coverage, population definition and experimental set up. How can these data sets be combined into a single source of data to mine in? This topic is addressed in chapter 3 and concerns the second step in the data mining process, the data step.

The fact that different classifiers produce similar results on the same data set is also making the point for going beyond mere performance evaluation. Evaluation methods should be used and developed that provide more of a diagnosis and characterization approach to evaluation rather than just measuring quality, a topic that is addressed in chapter 4 and 5. This fits the evaluation step in the data mining process, traditionally the fourth step straight after the modeling step, but as per the above the scope of this evaluation should not be constrained to the modeling only, a topic that chapter 4 is specifically dealing with. In the second case we did carry out a basic bias variance evaluation, but this was limited to comparing different modeling algorithms only. Just as in chapter 4 variance was a more important component than bias to explain differences across classifiers, and the experiments provided us with data to take a somewhat speculative attempt at estimating the intrinsic error, the error rate for the (theoretical) optimal classifier.

The final case introduces a real time automatic scene classifier for content-based video retrieval. In our envisioned approach end users like television archive documentalists, not image processing experts, build classifiers interactively, by simply indicating positive examples of a scene. To produce classifiers that are sufficiently reliable we have developed a procedure for generating problem specific data preprocessors. This approach has been successfully applied to various domains of video content analysis, such as content based video retrieval in television archives, automated sewer inspection, and porn filtering. In our opinion, in most circumstances the ultimate goal for data mining is to let end users create classifiers, primarily because it will be more scalable; a lot more classifiers can be created in much shorter time by a lot more users. In addition the resulting models can be of higher quality compared to purely data driven approaches, no matter how advanced the algorithms would be, because experts can inject domain knowledge by identifying relevant preprocessors. In terms of the data mining process, this case is more concerned with changing the agent executing the end to end process from data miner to domain expert.

In summary, given the apparent limited impact of modeling methods given a prepared data set, these cases demonstrate the importance of developing generally applicable methodology and tools to improve all steps in the process beyond modeling, starting with objective formulation, data collection and preparation to evaluation and deployment, which also opens up opportunities for end to end process automation.

In the data fusion chapter (chapter 3) we start by discussing how the information explosion provides not just opportunities but also barriers to the application of data mining, and position data fusion as a potential solution. The initial set of publications this chapter is based on were the first papers introducing data fusion to the data mining community. It was also the first reported case within which data fusion is used for a standard task in data preparation for data mining, enriching data sets so to build improved predictive models. This is illustrated with a case from database marketing. So far research literature on data fusion only covered fusion of surveys, not fusion of a customer database with a survey.

Despite its difficulties and pitfalls, the application of data fusion can increase the value of data mining, because there is more integrated data to mine. Data mining algorithms can also be used to perform fusions, but publications on methods other than the standard statistical matching approach are rare. Therefore we think that data fusion is an interesting topic for knowledge discovery and data mining research. This can also include research on automating the fusion process itself, including quality evaluation, and developing mining algorithms that automatically and appropriately deal with uncertainty in the data that is the result of the fusion.

In chapter 4 we have presented an analysis of the results of a data mining competition, which can be seen as a large scale, real world experiment to study data mining in the wild, rather than under laboratory conditions. The data used is noisy, participants worked under time and competitive pressure and rewards were offered for the best solutions. Participants were free to define their own data mining approach, including the translation of the business goal into mining objectives, data preparation, the choice of models to be used and the extent of experimentation and evaluation. This has resulted in a large spread in the performance for the prediction task in the competition, in contrast to the limited differences that were observed in chapter 2 when comparing classifiers on a given data set with no further data preparation. This provides further support for our claim that the other steps in the process (objective and approach, data preparation, evaluation) are key in explaining real world data mining results.

To analyze the causes for the spread in results in more detail we used bias variance analysis. Bias variance is not just a simple quality measurement method, it provides a diagnosis of what the most important sources of error are, bias or variance, and the strategies to minimize bias typically have an opposite effect on variance. Bias variance analysis is usually only applied for characterizing modeling algorithms, but in this chapter we have used it as a framework to evaluate all steps in the data mining process.

A lot of data mining research is concerned with developing algorithms that can model increasingly complex non linear relationships, and separate classes with intricate decision boundaries, in other words methods that reduce model representation bias. However, as some of the participants discovered, simpler methods work better

than complex ones and results may only get worse through extensive experimentation. It turns out that for this problem, as for many real world problems, variance is a more important cause for differences in performance. Even more important than the choice of classifier however is the preparation done in the data step. The distribution of predictive power across predictors is very skewed for this problem and many predictors are intercorrelated, so carrying out proper attribute selection is a more important factor than the choice of classifier. Also extensive experimentation and evaluation while keeping only the 'best' results can be dangerous. All the steps in the data mining process run the risk of increasing the variance error, all the way to the methods of model evaluation. In general, if it can't be ruled out that a problem is high variance, first simplify the data through data preparation and build a small number of simple (i.e. low variance, high bias) models first, as these are most robust.

A diagnosis method such as bias variance has proven to be very useful for in depth meta analysis of real world learning. In terms of future research the bias variance method itself can be improved more, as there is no single universally accepted measure yet for zero one loss, and for other selected loss functions and evaluation metrics definitions are lacking. Procedures could be developed at any point in the process that explicitly minimize either bias or variance rather than overall error. Also alternative diagnosis methods could be developed in addition to bias variance. In summary, model diagnosis is a very relevant research area, where there is still a lot to be gained.

Chapter 5 discusses an approach to benchmarking and profiling a novel algorithm, based on the example of the AIRS algorithm. AIRS is a so called artificial immune system or immunocomputing algorithm, the class of mining algorithms inspired by the immune system. The natural immune system is sometimes called the second brain given its capabilities to remember past intrusions and learn to recognize new ones, even if these vary from what was encountered before. AIRS is one of only two existing immunocomputing algorithms targeting the most common data mining task, prediction. We picked AIRS because as a new biologically inspired algorithm it may be at the risk of being an overhyped method, whereas one of our goals was to show there is no such thing as a free lunch.

This was the first basic benchmarking study of AIRS that compared AIRS across a wide variety of data sets and algorithms, using a consistent experimental set up rather than referring to benchmark results from literature. In contrast to earlier claims, and line what we expected, we find no evidence that AIRS consistently outperforms other algorithms. However, AIRS provides stable, near average results so it can safely be added to the data miners toolbox.

As discussed, the value of performance benchmarking is limited, because it doesn't provide a diagnosis of the sources of error (chapter 2 and 4) and is often limited to the modeling step alone (in contrast to the approach in chapter 4). Another issue is that, given the no free lunch theorem, there will be no methods that

significantly outperform all other methods across all problem domains. Basic benchmarking will very likely not prove that the novel method will beat all others, it can merely provide smoke test results that the novel algorithm does not perform significantly worse, as a minimal test to pass. Given this it will be more interesting to characterize the novel algorithm, for instance to determine on what kind of subsets of problems does a method perform better than average, and to what other algorithms does the novel algorithm compare in terms of patterns of performance over data sets. We introduced the methodological concept of algorithm profiling for this, to be used in addition to basic benchmarking, to evaluate the performance of novel algorithms.

To better identify the type of problems and data that fit AIRS we carried out experiments to compare the learning curves of methods by measuring accuracy over increasing data set sizes. The AIRS example results in a more or less standard curve for AIRS, steeper than multi layer perceptrons but flatter than nearest neighbor. We also explored a variety of methods for computing empirical algorithm similarity, based on performance patterns over data sets. The similarity measurements experiments confirm that AIRS behaves similar to nearest neighbor based methods, as can be expected from theoretical comparison.

AIRS was not the main topic of chapter 5, it was just used as an example to make the case for more research into algorithm profiling methods. Even proper basic benchmarking is often lacking for novel methods, but profiling is required for a deeper understanding of how the algorithm behaves and to what problems it should be applied. This can also include model diagnosis methods such as bias variance that will give the data miner more insight in the source of error, and thus the strategies to understand and improve the models (see chapter 4). Additional model profiling methods can be developed and applied to characterize a novel modeling algorithm in the context of a problem domain and the existing toolkit of algorithms available. In summary, diagnosis and profiling are interesting, original and attractive areas for further research.

This thesis has discussed a broad range of issues from quite a few angles. However, we aim to have provided a small number of consistent key messages. First and foremost we want to emphasize the importance and relevance to study data mining as an end to end process, rather than limit research to developing new modeling algorithms. The steps beyond the modeling step in the process are key, and methodologies and tools can be developed that apply not just to a single problem, but to a problem domain or even in general. Data fusion, model diagnosis and profiling are examples of these kind of tools. Taking an end to end view, and providing tools for all phases, will enable key steps forward such as end to end process automation, linking data mining to action to improve deployment and putting data mining in the hands of the end user, the domain expert rather than the data mining expert. These will be key factors in further scaling up to widespread application of data mining.

Bibliography

Aha, D. W. (1992), Generalizing from case studies: a case study, *in* 'Proceedings of the ninth international workshop on Machine learning (ML1992)', Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 1–10.

Baatenburg de Jong, R. J., Hermans, J., Molenaar, J., Briaire, J. & le Cessie, S. (2001), 'Prediction of survival in patients with head and neck cancer', *Head and Neck* **23**(9), 718–724.

Baker, K., Harris, P. & O'Brien, J. (1989), 'Data fusion: An appraisal and experimental evaluation', *Journal of the Market Research Society* **31**(2), 152–212.

Barr, R. & Turner, J. (1978), A new, linear programming approach to microdata file merging, *in* '1978 Compendium of Tax Research', Office of Tax Analysis.

Bei, Y., Belmamoune, M. & Verbeek, F. J. (2006), Ontology and image semantics in multimodal imaging: submission and retrieval, *in* S. Santini, R. Schettini & T. Gevers, eds, 'Proc. of SPIE Internet Imaging VII', Vol. 6061, SPIE, p. 60610C.

Belongie, S., Carson, C., Greenspan, H. & Malik, J. (1997), Recognition of images in large databases using a learning framework, Technical Report CSD-97-939, University of California at Berkeley.

Berka, P. (1999), Workshop notes on Discovery Challenge PKDD-99, Technical report, Laboratory of Intelligent Systems, University of Economics, Prague.

Berlin, B. & Kay, P. (1969), *Basic color terms: Their universals and evolution*, Berkeley: University of California Press.

Berry, M. J. & Linoff, G. (1997), *Data mining techniques for marketing, sales, and customer support*, John Wiley & Sons, New York, NY.

Bishop, C. M. (1995), *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK.

Blake, C. & Merz, C. (1998), 'UCI repository of machine learning databases'.

Borg, I. & Groenen, P. (2005), *Modern Multidimensional Scaling: theory and applications (2nd ed.)*, Springer Series in Statistics, Springer Verlag New York.

Bose, I., Reese, A. J., Ory, J. J., Janbon, G. & Doering, T. L. (2003), 'A yeast under cover: the capsule of Cryptococcus neoformans', *Eukaryot. Cell* **2**(4), 655–663.

Breiman, L. (1996), Bias, variance, and arcing classifiers, Technical report, Statistics Department, University of California.

Brox, T., Weickert, J., Burgeth, B. & Mrázek, P. (2006), 'Nonlinear structure tensors', *Image and Vision Computing* **24**(1), 41–55.

Budd, E. (1971), 'The creation of a microdata file for estimating the size distribution of income', *Review of Income and Wealth* **17**, 317–333.

Campbell, N. W., Mackeown, W. P. J., Thomas, B. T. & Troscianko, T. (1996), The automatic classification of outdoor images, *in* 'Proceedings of the International Conference on Engineering Applications of Neural Networks', Systems Engineering Association, pp. 339–342.

Campbell, N. W., Mackeown, W. P. J., Thomas, B. T. & Troscianko, T. (1997), 'Interpreting image databases by region classification', *Pattern Recognition* **30**(4), 555–563.

Carson, C., Belongie, S., Greenspan, H. & Malik, J. (2002), 'Blobworld: Image segmentation using expectation-maximization and its application to image querying', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(8), 1026–1038.

Carter, J. (2000), 'The immune system as a model for pattern recognition and classification', *Journal of the American Medical Informatics Association* **7**(1), 28–41.

Casadevall, A. & Perfect, J. R. (1998), *Cryptococcus neoformans*, ASM Press, Washington.

Castro, L. D. & Timmis, J. (2002), *Artificial Immune Systems: a New Computational Intelligence Approach*, Springer Verlag.

Castro, L. D. & von Zuben, F. (2000), The clonal selection algorithm with engineering applications, *in* D. Whitley, D. Goldberg, E. CantuPaz, L. Spector, I. Parmee & H. Beyer, eds, 'Workshop Proceedings of GECCO 2000, Workshop on Artificial Immune Systems and Their Applications', pp. 36–37.

Chandrinos, K. V., Androutsopoulos, I., Paliouras, G. & Spyropoulos, C. D. (2000), Automatic web rating: Filtering obscene content on the web, *in* J. Borbinha & T. Baker, eds, 'Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries', pp. 403–406.

Chang, Y. C. & Kwon-Chung, K. J. (1994), 'Complementation of a capsule-deficiency mutation of Cryptococcus neoformans restores its virulence', *Mol. Cell. Biol* **14**(7), 4912–4919.

Chapman, P., Clinton, J., Khabaza, T., Reinartz, T. & Wirth, R. (1999), The CRISP-DM process model, Technical report, Crisp Consortium. http://www.crisp-dm.org/.

Contrino, H., McGuckin, N. & Banks, D. (2000), Exploring the full continuum of travel: Data fusion by recursive partitioning regression, *in* 'International Association of Travel Behavior Research Conference (IATBR)'.

Cox, I. J., Miller, M. L., Minka, T. P. & Papathomas, T. V. (2000), 'The bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments', *IEEE Transactions on Image Processing* **9**(1), 20–37.

Cubo, Ó., Robles, V., Segovia, J. & Ruiz, E. M. (2005), Using genetic algorithms to improve accuracy of economical indexes prediction, *in* '6th International Symposium on Intelligent Data Analysis (IDA 2005)', pp. 57–65.

de Ruiter, M. (1999), Bayesian classification in data mining: theory and practice, Master's thesis, BWI, Free University of Amsterdam, The Netherlands.

Derefeldt, G. & Swartling, T. (1995), 'Colour concept retrieval by free colour naming: Identification of up to 30 colours without training', *Displays* **16**(2), 69–77.

Derefeldt, G., Swartling, T., Berggrund, U. & Bodrogi, P. (2004), 'Cognitive color', *Color Research & Application* **29**(1), 7–19.

Domingos, P. (1997), 'The role of Occam's Razor in knowledge discovery', *Data Mining and Knowledge Discovery* **3**, 409–425.

Domingos, P. (2000), A unified bias-variance decomposition and its applications, *in* 'Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)', Morgan Kaufmann, CA, pp. 231–238.

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

D'Orazio, M., Zio, M. D. & Scanu, M. (2006), *Statistical Matching: Theory and Practice*, Wiley.

Dykstra, M. A., Friedman, L. & Murphy, J. W. (1977), 'Capsule size of Cryptococcus neoformans: control and relationship to virulence', *Infect. Immun.* **16**(1), 129–135.

Elkan, C. (2001), Magical thinking in data mining: Lessons from CoIL Challenge 2000, *in* 'Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining (KDD 2001)', pp. 426–431.

Engels, R., Lindner, G. & Studer, R. (1997), A guided tour through the data mining jungle, *in* 'Proceedings of the 3rd International Conference on Knowledge Discovery in Databases (KDD 1997)', Newport Beach, CA.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P. & Uthurusamy, R. (1996), *Advances in Knowlege Discovery and Data Mining*, MIT Press, Cambridge, Mass.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D. & Yanker, P. (1995), 'Query by Image and Video Content: The QBIC system', *IEEE Computer* **28**(9), 23–32.

Flores, G. A. & Albacea, E. A. (2007), A genetic algorithm for constrained statistical matching, *in* '10th National Convention on Statistics (NCS), Manila, Phillipines'.

Forsyth, D. A. & Ponse, J. (2002), *Computer Vision: A modern approach*, Pearson Education, Inc., Upper Saddle River, New Jerssey, U.S.A.

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

Fritzke, B. (1994), 'Growing cell structures — a self-organzing network for unsupervised and supervised learning', *Neural Networks* **7**, 1441–1460.

Fritzke, B. (1995), A growing neural gas network learns topologies, *in* 'Advances in Neural Information Processing Systems 7', MIT Press, pp. 625–632.

Fung, C. Y. & Loe, K.-F. (1999*a*), Learning primitive and scene semantics of images for classification and retrieval, *in* 'Proceedings of the 7th ACM International Conference on Multimedia '99', Vol. 2, ACM, Orlando, Florida, USA, pp. 9–12.

Fung, C. Y. & Loe, K.-F. (1999*b*), A new approach for image classification and retrieval, *in* 'Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval', ACM, pp. 301–302.

Geman, S., Bienenstock, E. & Doursat, R. (1992), 'Neural networks and the bias/variance dilemma', *Neural Computation* **4**, 1–58.

Gevers, T. & Smeulders, A. W. M. (1999), 'Color based object recognition', *Pattern Recognition* **32**(3), 453–464.

Gevers, T. & Smeulders, A. W. M. (2000), 'Pictoseek: combining color and shape invariant features for image retrieval', *IEEE Transactions on Image Processing* **9**(1), 102–119.

Gibson, J. (1979), *The Ecological Approach to Visual Perception*, Houghton Mifflin, Boston.

Goldstone, R. (1995), 'Effects of categorization on color perception', *Psychological Science* **5**(6), 298–304.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J., Caligiuri, M., Bloomfield, C. D. & Lander, E. S. (1999), 'Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring', *Science* **286**(5439), 531–537.

Golubev, W. L. & Manukyan, A. R. (1979), 'Capsule formation by a saprophytic yeast', *Mikrobiologiya* **48**(314).

Gonzales, R. C. & Woods, R. E. (1993), *Digital Image Processing*, Prentice Hall, New Jersey.

Goodman, D., Boggess, L. & Watkins, A. (2002), Artificial immune system classification of multiple-class problems, *in* 'Artificial Neural Networks in Engineering (ANNIE 2002)'.

Goodman, D., Boggess, L. & Watkins, A. (2003), An investigation into the source of power for AIRS, an artificial immune classification system, *in* 'Proceedings IJCNN 2003'.

Gorkani, M. M. & Picard, R. W. (1994), Texture orientation for sorting photos at a glance, *in* 'Proceedings of the International Conference on Pattern Recognition', Vol. 1, pp. 459–464.

Gusfield, D. & Irving, R. W. (1989), *The Stable Marriage Problem: structure and algorithms*, MIT Press.

Guyon, I. & Elissee, A. (2003), 'An introduction to variable and feature selection', *Journal of Machine Learning Research* **3**, 1157–1182.

Hall, M. A. (1999), Correlation-based Feature Subset Selection for Machine Learning, PhD thesis, University of Waikato.

Hall, M. A. & Holmes, G. (2003), 'Benchmarking attribute selection techniques for discrete class data mining', *IEEE Transactions on knowledge and Data Engineering* **15**(6), 1437–1447.

Hamaker, J. & Watkins, A. (2003), *Artificial Immune Recognition System (AIRS), Java source code*.

Harrell, F. (2001), *Regression Modeling Strategies: with Applications to Linear Models, Logistic Regression and Survival Analysis*, Springer Series in Statistics, Springer Verlag.

Holte, R. (1993), 'Very simple classification rules perform well on most commonly used datasets', *Machine Learning* **11**, 63–91.

Hu, M. (1962), 'Visual pattern recognition by moment invariants', *IRE Trans. on Information Theory* **8**(2), 179–187.

IBM Research (2005), MARVEL: Multimedia analysis and retrieval system, Technical report, Intelligent Information Management Dept., IBM T. J. Watson Research Center.

Israël, M. (1999), Parabot: Text and image classification for the internet, internal report, Sentient Machine Research.

Israel, M., van den Broek, E. L., van der Putten, P. & den Uyl, M. (2004*a*), Real time automatic scene classification, *in* 'Demonstration Paper at BNAIC 2004', Groningen, The Netherlands.

Israël, M., van den Broek, E. L., van der Putten, P. & den Uyl, M. J. (2004*b*), Automating the construction of scene classifiers for content-based video retrieval, *in* L. Khan & V. A. Petrushin, eds, 'Proceedings of the Fifth International Workshop on Multimedia DataMining (MDM/KDD'04)', Seattle, WA, USA, pp. 38–47.

Israël, M., van den Broek, E. L., van der Putten, P. & den Uyl, M. J. (2006), Visual alphabets: Video classification by end users, *in* V. A. Petrushin & L. Khan, eds, 'Multimedia Data Mining and Knowledge Discovery', Springer, chapter 10, pp. 185–203.

Jähne, B. (1997), *Practical Handbook on Image Processing for Scientific Applications*, CRC Press.

James, G. M. (2003), 'Variance and bias for general loss functions', *Machine Learning* **51**, 115–135.

Janbon, G. (2004), 'Cryptococcus neoformans capsule biosynthesis and regulation', *FEMS Yeast Res* **4**(8), 765–771.

Jephcott, J. & Bock, T. (1998), 'The application and validation of data fusion', *Journal of the Market Research Society* **40**(3), 185–205.

Kamakura, W. & Wedel, M. (1997), 'Statistical data fusion for cross-tabulation', *Journal of Marketing Research* **34**(4), 485–498.

Kay, P. (1999), 'Color', *Journal of Linguistic Anthropology* **1**, 29–32.

Kohavi, R. & John, G. (1997), 'Wrappers for feature subset selection', *Artificial Intelligence* **97**, 273–324.

Kohavi, R. & Wolpert, D. H. (1996), Bias plus variance decomposition for zero-one loss functions, *in* L. Saitta, ed., 'Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996)', Morgan Kaufmann, pp. 275–283.

Kohonen, T. (1982), 'Self-organized formation of topologically correct feature maps', *Biological Cybernetics* **43**, 59–69.

Kum, H. & Masterson, T. (2008), Statistical matching using propensity scores: Theory and application to the levy institute measure of economic well-being, Working paper no. 535, The Levy Economics Institute of Bard College.

Lin, T. & Zhang, H. (2000), Automatic video scene extraction by shot grouping, *in* 'Proceedings of the 15th International Conference on Pattern Recognition', Vol. 4, IEEE Computer Society, Barcelona, Spain, pp. 39–42.

Ling, C. X. & Li, C. (1998), Data mining for direct marketing: Problems and solutions, *in* 'Proceedings 4th International Conference on Knowledge Discovery in Databases (KDD 1998)', New York.

Lipson, P., Grimson, E. & Sinha, P. (1997), Configuration based scene classification and image indexing, *in* 'Proceedings of 16th IEEE Conference on Computer Vision and Pattern Recognition', IEEE Computer Society, pp. 1007–1013.

Little, R. & Rubin, D. (1986), *Statistical analysis with missing data*, John Wiley and Sons.

Littman, M. L. (1958), 'Capsule synthesis by Cryptococcus neoformans', *Trans NY Acad Sci* **20**(7), 623–648.

Littman, M. L. & Tsubura, E. (1959), 'Effect of degree of encapsulation upon virulence of Cryptococcus neoformans', *Proc. Soc. Exp. Biol. Med.* **101**, 773–777.

Liu, J., van der Putten, P., Hagen, F., Chen, X., Boekhout, T. & Verbeek, F. (2006), Detecting virulent cells of Cryptococcus neoformans yeast: Clustering experiments, *in* 'ICPR (1)', IEEE Computer Society, pp. 1112–1115.

Liu, X. & Kellam, P. (2003), Mining gene expression data, *in* D. J. C. Orengo & J. Thornton, eds, 'Bioinformatics: Genes, Proteins and Computers', BIOS Scientic Publishers, pp. 229–244.

Lonardi, S., Chen, J. Y. & Zaki, M. J., eds (2008), *Proceedings of the ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD 2008)*.

Maat, B. (2006), The need for fusing head and neck cancer data: can more data provide a better data mining model for predicting survivability of head and neck cancer patients?, Master's thesis, ICT in Business, Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands.

Martinetz, T. (1993), Competitive Hebbian learning rule forms perfectly topology preserving maps, *in* 'Proceedings ICANN-93 Amsterdam', pp. 427–434.

Martinetz, T. & Schulten, K. (1994), 'Topology representing networks', *Neural Networks* **7**(3), 507–522.

Marwah, G. & Boggess, L. (2002), Artificial immune systems for classification: Some issues, *in* '1st International Conference on Artificial Immune Systems', pp. 149–153.

McCulloch, W. & Pitts, W. (1943), 'A logical calculus of the ideas immanent in nervous activity', *Bulletin of Mathematical Biophysics* **5**, 115–133.

Meng, L., van der Putten, P. & Wang, H. (2005), A comprehensive benchmark of the artificial immune recognition system (AIRS), *in* 'Proceedings First International Conference on Advanced Data Mining and Applications (ADMA 2005), Wuhan, China', Vol. 3584 of *Lecture Notes in Computer Science*, Springer, pp. 575–582.

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* 'AAAI 1986', Morgan Kaufmann, Philadelphia, PA, pp. 1041–1047.

Minka, T. P. & Picard, R. W. (1996), Interactive learning using a "Society of Models", Technical Report 349, MIT Media Laboratory Perceptual Computing Section.

Moller, M. F. (1993), 'A scaled conjugate gradient algorithm for fast supervised learning', *Neural Networks* **6**(4), 525–533.

Nguyen, D. & Widrow, B. (1990), Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights, *in* 'IJCNN International Joint Conference on Neural Networks', Vol. 3, pp. 21–26.

Niblack, W., Barber, R., Equitz, W., Flickner, M., Glasman, E., Petkovic, D., Yanker, P. & Faloutos, C. (1993), The QBIC project: Querying images by content using color, texture, and shape, *in* W. Niblack, ed., 'Proceedings of Storage and Retrieval for Image and Video Databases', Vol. 1908, pp. 173–187.

Noll, P. (2009), Statistisches Matching mit Fuzzy Logic, PhD thesis, Philipps-Universität Marburg.

Noll, P. & Alpar, P. (2007), A methodology for statistical matching with fuzzy logic, *in* 'Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2007)', pp. 73–78.

O'Brien, S. (1991), 'The role of data fusion in actionable media targeting in the 1990s', *Marketing and Research Today* **19**, 15–22.

Oliva, A. & Torralba, A. (2001), 'Modeling the shape of the scene: A holistic representation of the spatial envelope', *International Journal of Computer Vision* **42**(3), 145–175.

Paass, G. (1986), Statistical match: Evaluation of existing procedures and improvements by using additional information, *in* G. Orcutt & K. Merz, eds, 'Microanalytic Simulation Models to Support Social and Financial Policy', Elsevier Science, pp. 401–422.

Paauwe, P., van der Putten, P. & van Wezel, M. C. (2007), DTMC: an actionable e-customer lifetime value model based on markov chains and decision trees, *in* 'Proceedings of the 9th International Conference on Electronic Commerce: The Wireless World of Electronic Commerce (ICEC 2007)', pp. 253–262.

Palm, C. (2004), 'Color texture classification by integrative co-occurrence matrices', *Pattern Recognition* **37**(5), 965–976.

Pei, J., Getoor, L. & de Keijzer, A., eds (2009), *First ACM SIGKDD Workshop on Knowledge Discovery from Uncertain Data, Paris, France, June 28*, ACM.

Picard, R. W. (1995), Light-years from Lena: video and image libraries of the future, *in* 'Proceedings of the 1995 International Conference on Image Processing', Vol. 1, pp. 310–313.

Picard, R. W. & Minka, T. P. (1995), 'Vision texture for annotation', *Multimedia Systems* **3**(1), 3–14.

Prasad, B., Gupta, S. & Biswas, K. (2001), Color and shape index for region-based image retrieval, *in* C. Arcelli, L. Cordella & G. S. di Baja, eds, 'Proceedings of 4th International Workshop on Visual Form', Springer Verlag, Capri, Italy, pp. 716–725.

Quinlan, J. R. (1986), 'Induction of decision trees', *Machine Learning* **1**, 81–106.

Quinlan, J. R. & Cameron-Jones, R. M. (1995), Oversearching and layered search in empirical learning, *in* 'IJCAI 1995', pp. 1019–1024.

Radner, D., Rich, A., Gonzalez, M., Jabine, T. & Muller, H. (1980), Report on exact and statistical matching techniques. statistical working paper 5, Technical report, Office of Federal Statistical Policy and Standards US DoC.

Raessler, S. (2002), *Statistical Matching: A Frequentist Theory, Practical Applications, and Alternative Bayesian Approaches*, Springer.

Ramon, J., Costa, F., Florencio, C. C. & Kok, J., eds (2008), *Statistical and Relational Learning in Bioinformatics (StReBio 2008) at ECML-PKDD 2008*.

Ratan, A. L. & Grimson, W. E. L. (1997), Training templates for scene classication using a few examples, *in* 'Proceedings of the IEEE Workshop on Content-Based Analysis of Images and Video Libraries', pp. 90–97.

Redfield, S., Nechyba, M., Harris, J. G. & Arroyo, A. A. (2001), Efficient object recognition using color, *in* R. Roberts, ed., 'Proceedings of the Florida Conference on Recent Advances in Robotics', Tallahassee, Florida.

Rivera, J., Feldmesser, M., Cammer, M. & Casadevall, A. (1998), 'Organ-dependent variation of capsule thickness in Cryptococcus neoformans during experimental murine infection', *Infect. Immun.* **66**(10), 5027–5030.

Roberson, D., Davies, I. & Davidoff, J. (2000), 'Color categories are not universal: Replications and new evidence from a stone-age culture', *Journal of Experimental Psychology: General* **129**, 369–398.

Roberts, A. (1994), 'Media exposure and consumer purchasing: An improved data fusion technique', *Marketing And Research Today* **22**, 159–172.

Rodgers, W. L. (1984), 'An evaluation of statistical matching', *Journal of Business & Economic Statistics* **2**(1), 91–102.

Rosenfeld, A. (2001), 'From image analysis to computer vision: An annotated bibliography, 1955-1979', *Computer Vision and Image Understanding* **84**(2), 298–324.

Rubin, D. B. (1986), 'Statistical matching using file concatenation with adjusted weights and multiple imputations', *Journal of Business & Economic Statistics* **4**(1), 87–94.

Ruggles, N. & Ruggles, R. (1974), 'A strategy for merging and matching microdata sets', *Annals Of Social And Economic Measurement* **3**(2), 353–371.

Rumelhart, D. E. & McClelland, J. L. (1986), *Parallel Distributed Processing: explorations in the microstructure of cognition*, MIT Press, Cambridge, Mass.

Schettini, R., Ciocca, G. & Zuffi, S. (2001), A survey of methods for colour image indexing and retrieval in image databases, *in* R. Luo & L. MacDonald, eds, 'Color imaging science: Exploiting Digital media', J. Wiley.

Smith, J. R. & Chang, S.-F. (1995), Single color extraction and image query, *in* B. Liu, ed., 'Proceedings of the 2nd IEEE International Conference on Image Processing', IEEE Signal Processing Society, IEEE Press, pp. 528–531.

Smith, J. R. & Chang, S. F. (1997), Querying by color regions using the visualseek content-based visual query system, *in* M. T. Maybury, ed., 'Intelligent Multimedia Information Retrieval', The AAAI Press, chapter 2, pp. 23–42.

Smith, K. A., Chuan, S. & van der Putten, P. (2001), Determining the validity of clustering for data fusion, *in* A. Abraham & M. Köppen, eds, 'Proceedings Hybrid Information Systems, First International Workshop on Hybrid Intelligent Systems, Adelaide, Australia, December 11-12, 2001 (HIS 2001)', Advances in Soft Computing, Physica-Verlag, pp. 627–636.

Soares, C. & Brazdil, P. (2000), Zoomed ranking: Selection of classification algorithms based on relevant performance information, *in* 'Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD 2000)', Springer-Verlag, London, UK, pp. 126–135.

Soong, R. & de Montigny, M. (2003), Does fusion-on-the-fly really fly?, *in* 'ARF/ESOMAR Week of Audience Measurement'.

Soong, R. & de Montigny, M. (2004), No Free Lunch in data integration, *in* 'ARF/ESOMAR Week of Audience Measurement'.

Sun, Z. (2005), EQPD, a way to improve the accuracy of mining fused data?, Master's thesis, ICT in Business, Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands.

Szummer, M. & Picard, R. W. (1998), Indoor-outdoor image classification, *in* 'IEEE International Workshop on Content-Based Access of Image and Video Databases (CAIVD)', IEEE Computer Society, Bombay, India, pp. 42–51.

Tax, D. (2001), One-class classification; concept learning in the absence of counter-examples, PhD thesis, Delft University of Technology.

Timmis, J. & Neal, M. (2001), 'A resource limited artificial immune system. knowledge based systems', *Knowledge Based Systems* **14**(3/4), 121–130.

Tsamardinos, I. & Aliferis, C. (2003), Towards principled feature selection: Relevancy, filters and wrappers, *in* 'Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics'.

van Balen, R., Koelma, D., ten Kate, T. K., Mosterd, B. & Smeulders, A. (1993), Scilimage: A multi-layered environment for use and development of image processing software, *in* H. Christensen, ed., 'Experimental environments for computer vision and image processing', World Scientific, Singapore.

van den Broek, E. L., Kisters, P. M. F. & Vuurpijl, L. G. (2005), 'Content-based image retrieval benchmarking: Utilizing color categories and color distributions', *Journal of Imaging Science and Technology* **49**(3), 293–301.

van den Broek, E. L., Kok, T., Schouten, T. E. & Hoenkamp, E. (2006), 'Multimedia for Art ReTrieval (M4ART)', *Proceedings of SPIE (Multimedia Content Analysis, Management, and Retrieval)* **6073**, 60730Z.

van den Broek, E. L. & van Rikxoort, E. M. (2005), 'Parallel-sequential texture analysis', *Lecture Notes in Computer Science (Advances in Pattern Recognition)* **3687**, 532–541.

van den Broek, E. L., van Rikxoort, E. M., Kok, T. & Schouten, T. E. (2006), M-hints: Mimicking humans in texture sorting, *in* B. E. Rogowitz, T. N. Pappas & S. J. Daly, eds, 'Human Vision and Electronic Imaging XI', SPIE, the International Society for Optical Engineering. Proceedings of SPIE, vol. 6057.

van den Broek, E. L., van Rikxoort, E. M. & Schouten, T. E. (2005), 'Human-centered object-based image retrieval', *Lecture Notes in Computer Science (Advances in Pattern Recognition)* **3687**, 492–501.

van der Putten, P. (1996), Utilizing the topology preserving property of self-organizing maps for classification, Master's thesis, Cognitive Artificial Intelligence, Utrecht University, The Netherlands.

van der Putten, P. (1999*a*), Data mining in direct marketing databases, *in* W. Baets, ed., 'Complexity and Management: A Collection of Essays', World Scientific Publishers, Singapore, pp. 97–111.

van der Putten, P. (1999*b*), 'Datamining in bedrijf', *Informatie en Informatiebeleid* **17**(3), 15–25.

van der Putten, P. (1999*c*), A datamining scenario for stimulating credit card usage by mining transaction data, *in* 'Benelearn 1999', Maastricht, The Netherlands, pp. 105–111.

van der Putten, P. (1999*d*), Promoting credit card usage by mining transaction data, *in* P. Berka, ed., 'Workshop notes on Discovery Challenge PKDD-99', Laboratory of Intelligent Systems, University of Economics, Prague.

van der Putten, P. (1999*e*), 'Vicar Video Navigator: Content based video search engines become a reality', *Broadcast Hardware International, IBC edition* **80**, 51–53.

van der Putten, P. (2000*a*), Data fusion: A way to provide more data to mine in?, *in* 'Proceedings 12th Belgian-Dutch Artificial Intelligence (BNAIC 2000)', pp. 133–140.

van der Putten, P. (2000*b*), Data fusion for data mining: a problem statement, *in* 'Coil Seminar 2000', Chios, Greece, pp. 96–101.

van der Putten, P. (2002*a*), Advertising strategy discovery, *in* J. Meij, ed., 'Dealing with the Data Flood: Mining Data, Text and Multimedia', STT 65, STT The Hague, pp. 247–261.

van der Putten, P. (2002*b*), Analytical customer relationship management for insurance policy prospects, *in* J. Meij, ed., 'Dealing with the Data Flood: Mining Data, Text and Multimedia', STT 65, STT The Hague, pp. 293–297.

van der Putten, P. (2002*c*), 'Broodnodige intelligentie voor CRM', *Database Magazine* (7), 6–9.

van der Putten, P. (2009), 'Data mining is dood, lang leve decisioning', *Database Magazine* (3), 16–21.

van der Putten, P., Bertens, L., Liu, J., Hagen, F., Boekhout, T. & Verbeek, F. J. (2007), Classification of yeast cells from image features to evaluate pathogen conditions, *in* A. Hanjalic, R. Schettini & N. Sebe, eds, 'Multimedia Content Access: Algorithms and Systems', Vol. 6506:1, Proceedings of the SPIE.

van der Putten, P. & Kok, J. N. (2005), Data mining and knowledge discovery, *in* R. J. B. de Jong, ed., 'Prognosis in Head and Neck Cancer', Taylor and Francis, chapter 18, pp. 303–314.

van der Putten, P., Kok, J. N. & Gupta, A. (2002*a*), Data fusion through statistical matching, Technical Report Working Paper No. 4342-02, MIT Sloan School of Management, Cambridge, MA.

van der Putten, P., Kok, J. N. & Gupta, A. (2002*b*), Why the information explosion can be bad for data mining, and how data fusion provides a way out, *in* R. L. Grossman, J. Han, V. Kumar, H. Mannila & R. Motwani, eds, 'Second SIAM International Conference on Data Mining (SDM 2002)', SIAM, pp. 128–138.

van der Putten, P., Koudijs, A. & Walker, R. (2004), Basel II compliant credit risk management: the OMEGA case, *in* J. van den Berg & U. Kaymak, eds, '2nd EUNITE Workshop on Smart Adaptive Systems in Finance: Intelligent Risk Analysis and Management', Rotterdam, The Netherlands, pp. 17–19.

van der Putten, P., Koudijs, A. & Walker, R. (2006), A decision management approach to Basel II compliant credit risk management, *in* R. Ghani & C. Soares, eds, 'Workshop Data Mining for Business Applications (DMBA 2006) at KDD 2006', Philadelphia, Pennsylvania, USA, pp. 71–75.

van der Putten, P. & Meng, L. (2005), Benchmarking artificial immune systems, *in* K. Verbeeck, K. Tuyls, A. Nowé, B. Manderick & B. Kuijpers, eds, 'Proceedings of the Seventeenth Belgium-Netherlands Conference on Artificial Intelligence

(BNAIC 2005), Brussels, Belgium, October 17-18, 2005', Koninklijke Vlaamse Academie van Belgie voor Wetenschappen en Kunsten, pp. 393–394.

van der Putten, P., Meng, L. & Kok, J. N. (2008), Profiling novel classification algorithms: Artificial immune systems, *in* '7th IEEE International Conference on Cybernetic Intelligent Systems 2008 (CIS2008)', London, United Kingdom.

van der Putten, P., Ramaekers, M., den Uyl, M. & Kok, J. N. (2002), A process model for a data fusion factory, *in* 'Proceedings of the 14th Belgium/Netherlands Conference on Artificial Intelligence (BNAIC 2002)', Leuven, Belgium, pp. 251–258.

van der Putten, P. & van Someren, M. (1999), The Benelearn 1999 competition, Technical report, Sociaal Wetenschappelijke Informatica, Universiteit van Amsterdam.

van der Putten, P. & van Someren, M. (2000), CoiL Challenge 2000: the Insurance Company Case, Technical Report 2000-09, Leiden Institute of Advanced Computer Science, Universiteit van Leiden.

van der Putten, P. & van Someren, M. (2004), 'A bias-variance analysis of a real world learning problem: The CoIL Challenge 2000', *Machine Learning* **57**(1-2), 177–195.

van Hattum, P. & Hoijtink, H. (2008), 'The proof of the pudding is in the eating. data fusion: an application in marketing', *Journal of Database Marketing and Customer Strategy Management* **15**(4), 267–284.

van Hattum, P. & Hoijtink, H. (2009), 'Improving your sales with data fusion', *Journal of Database Marketing and Customer Strategy Management* **16**(1), 7–14.

van Pelt, X. (2001), The Fusion Factory: A constrained data fusion approach, Master's thesis, Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands.

van Rikxoort, E. M., van den Broek, E. L. & Schouten, T. E. (2005), 'Mimicking human texture classification', *Proceedings of SPIE (Human Vision and Electronic Imaging X)* **5666**, 215–226.

Verbeek, F. J. (1995), Three Dimensional reconstruction of biological objects from serial sections including deformation correction, PhD thesis, Delft University of Technology, Delft, The Netherlands.

Verbeek, F. J. (1999), Theory and practice of 3d-reconstructions from serial sections, *in* R. Baldock & J. Graham, eds, 'in: Image Processing, A Practical Approach', Oxford University Press, Oxford.

Vilalta, R. & Drissi, Y. (2002), 'A perspective view and survey of meta-learning', *Artif. Intell. Rev.* **18**(2), 77–95.

Wagenaar, E. (1997), *Data Mining in Marketing Databases*, DMSA, Amsterdam, NL. In Dutch. In cooperation with Marten den Uyl en Peter van der Putten.

Wang, J. Z. (2001), *Integrated region-based image retrieval*, Boston: Kluwer Academic Publishers.

Watkins, A. (2001), AIRS: A resource limited artificial immune classifier, Master's thesis, Department of Computer Science. Mississippi State University.

Watkins, A., Timmis, J. & Boggess, L. (2004), 'Artificial immune recognition system (AIRS): An immune-inspired supervised learning algorithm', *Genetic Programming and Evolvable Machines* **5**(3), 291–317.

Wirth, R., Shearer, C., Grimmer, U., Reinartz, T., Schlösser, J., Breitner, C., Engels, R. & Lindner, G. (1997), Process-oriented tool support for KDD, *in* 'Proceedings of the 1st European Symposium on Principles of Data Mining and Knowledge Discovery', Trondheim, Norway.

Witten, I. & Frank, E. (2000), *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann Publishers, San Francisco.

Wolpert, D. & MacReady, W. (1995), No free lunch theorems for search, Technical Report SFI-TR-95-02-010, The Santa Fe Institute. ftp://ftp.santafe.edu/pub/wgm/nfl.ps.

Wu, S., Rahman, M. K. M. & Chow, T. W. S. (2005), 'Content-based image retrieval using growing hierarchical self-organizing quadtree map', *Pattern Recognition* **38**(5), 707–722.

Zaragoza, O. & Casadevall, A. (2004), 'Experimental modulation of capsule size in Cryptococcus neoformans', *Biol. Proced, Online* **6**(1).

Zaragoza, O., Fries, B. C. & Casadevall, A. (2003), 'Induction of capsule growth in Cryptococcus neoformans by mammalian serum and CO_2', *Infect. Immun.* **71**(11), 6155–6164.

Chapter 7

Samenvatting

Het zit in de menselijke natuur om patronen te ontdekken in data. Mensen en dieren leren van de omgeving en bouwen zo kennis en intelligentie op. Methoden om theorie uit data af te leiden vormen de basis van empirische wetenschappen, en sinds de sterke opkomst hiervan zijn deze methoden zelf ook onderwerp van onderzoek. Rond het moment van de geboorte van de computer circa 65 jaar geleden werden de eerste papers al geschreven over hoe computers zouden kunnen leren aan de hand van data. Later zijn hiervoor nieuwe termen geïntroduceerd: data mining of knowledge discovery in databases, het ontdekken van interessante, betekenisvolle en nuttige patronen verborgen in data.

Toepassingen voor de gewone consument, burger of werknemer, voor dagelijks gebruik, hebben echter lang op zich laten wachten. Zo'n tien jaar geleden waren data mining toepassingen met name nog te vinden op universiteiten en onderzoeksinstituten, maar tegenwoordig wordt het op veel bredere schaal in de praktijk ingezet. Voorbeelden zijn internet zoekmachines, fraude controle bij credit card betalingen, medische diagnose en advies systemen voor doktoren en het slim en gepersonaliseerd sturen van klant interacties.

De titel van deze dissertatie, 'On Data Mining in Context: Cases, Evaluation and Fusion', heeft een dubbele betekenis. Het onderzoek wordt gemotiveerd en gedreven door de context van praktijktoepassingen, met name op bedrijfs- en biomedisch gebied, met als doel een bijdrage te leveren aan het nog grootschaliger en beter toepasbaar maken van data mining in de praktijk. Dit wil niet zeggen dat het onderzoek beperkt is tot cases, we streven er naar om interessante gebieden voor onderzoek te identificeren en illustratieve methoden en technieken te ontwikkelen die toepasbaar zijn op meerdere problemen of probleemdomeinen.

Het woord context refereert ook aan de stappen in een data mining proces. Standaard wordt dit proces ingedeeld in het vaststellen van een doelstelling en het

doorvertalen naar een data mining aanpak, het verzamelen en voorbewerken van benodigde data, het modelleren zelf waarbij met verschillende methoden patronen en verbanden worden ontdekt, de evaluatie van de kwaliteit van deze modellen en als laatste de toepassing ervan op nieuwe gevallen. Veel data mining research concentreert zich op de kern modelleer stap, dit onderzoek richt zich echter op een aantal specifieke problemen in de context van stappen voor en na de model stap (data en evaluatie) en het data mining proces als geheel, gegeven dat deze fasen met name voor praktijktoepassingen van groot belang kunnen zijn. Dit is een ambitieuze en brede doelstelling, in de aanpak kiezen we dan ook voor een variëteit van specifieke deelonderzoeken die passen binnen dit overkoepelende thema.

Als achtergrond worden er in hoofdstuk 2 een aantal praktijk cases besproken op het gebied van marketing, medische adviessystemen en beeldverwerking voor biologisch onderzoek, inhoud gebaseerd zoeken in televisiearchieven en internet content filtering. Een samenvatting van de cases kan gevonden worden in hoofdstuk 1 en aan het eind van hoofdstuk 2, maar een van de belangrijke lessen is dat de impact van de modelleerstap alleen op het data mining eindresultaat beperkt is; data mining in de praktijk behelst in ieder geval meer dan het toepassen van een algoritme op een data set. Dit levert verdere motivatie op voor de relevantie van het onderzoeksthema.

Hoofdstuk 3 gaat over een probleem uit de data stap in het data mining proces, het zogenaamde data fusie probleem. De meeste data mining algoritmen veronderstellen dat een enkele data set gebruikt wordt, maar in de praktijk kan informatie verspreid zijn over meerdere bronnen. Laten we het voorbeeld nemen van een klant. Een typisch probleem is dat informatie over de klant vaak te vinden is in meerdere tabellen in een onderliggende relationele database, terwijl de meeste standaard data mining algoritmen een plat record per klant verwachten, een rij per klant in een tabel met de meest belangrijke velden als kolommen. Informatie over de klant wordt dan verzameld middels database operaties zoals joins op unieke identifiers en aggegraties. Als matchende identifiers missen zijn er ook algoritmen om een meest waarschijnlijke match te maken.

Een ander probleem is hoe informatie over verschillende entiteiten met elkaar gecombineerd kan worden, in dit voorbeeld hoe voor een gegeven klant informatie van andere klanten gebruikt kan worden om het klantrecord te verrijken. Dit wordt ook wel data fusie genoemd en wordt hoofdzakelijk ingezet voor marketing, media en beleidseconomische onderzoeken, waarbij voor subgroepen van de respondenten verschillende vragen worden gesteld, en data fusie ingezet wordt voor het invullen van de ontbrekende vragen. Het gefuseerde bestand wordt dan gebruikt voor verdere data analyse, vaak vrij standaard statistisch van aard zoals kruistabellen e.d.

Dit hoofdstuk is gebaseerd op een aantal papers die de eerste publicaties zijn over data fusie voor een mainstream data mining publiek. Data fusie is naar onze mening een interessant nieuw onderwerp voor data miners omdat het barrières wegneemt voor de toepassing van data mining, en data mining voorspelalgoritmen ook ingezet

kunnen worden voor het uitvoeren van de fusie zelf. Om de relevantie voor data miners te verhogen, hebben we onderzocht of het mogelijk is dat data fusie tot betere resultaten kan leiden voor een klassieke data mining taak, namelijk voorspellen, in plaats van standaard data analyse. Een bestand is verrijkt middels fusie, en de modellen gebouwd op het verrijkte bestand zijn inderdaad beter dan de modellen gebouwd op het niet verrijkte bestand. Betere prestatie is echter niet gegarandeerd, en data fusie is een complex proces met beperkingen en valkuilen. We beschrijven een aantal van deze aspecten, en geven een overzicht van een data fusie proces model met stappen, vaak gebruikte methoden en technieken. Het ultieme doel van het proces model is om te komen tot een gestandaardiseerde, fabrieksmatige en geoptimaliseerde toepassing van data fusie, in een zogenaamde fusie fabriek.

Ten behoeve van het gebruikte marketing voorbeeld wordt een klantendatabase verrijkt met onderzoeksdata, door voor elke klant de verwachte antwoorden op het marktonderzoek te af te leiden, gegeven de onderzoeksrespondenten die het meest lijken op een gegeven klant. Vervolgens wordt onderzocht of we beter de kans op bezit van een credit card kunnen voorspellen met behulp van verrijkte data. Vanuit marketing perspectief is dit een van de eerste onderzoeken waarbij fusie niet wordt ingezet om marktonderzoeken te fuseren, maar om een klantendatabase te verrijken met onderzoeksdata, en zo een brug te slaan tussen marktonderzoek en direct marketing.

Zoals gezegd richten we ons op de stappen rondom de kernmodelleer stappen in het data mining proces, dus hoofdstuk 4 gaat met name over evaluatie, niet alleen van de modelleer stap, maar van het proces als geheel. Bij een aantal van de cases in hoofdstuk 2 bleek dat het gebruik van verschillende modelleermethoden, ceteris paribus, geen grote invloed heeft op de kwaliteit van het eindresultaat. Echter, in praktijktoepassingen kunnen de resultaten nogal verschillen tussen verscheidene start tot finish aanpakken van hetzelfde probleem. Experimenten onder gecontroleerde 'laboratoriumomstandigheden' kunnen tekort schieten bij het bestuderen van dit fenomeen. Om dit te onderzoeken is er een veldexperiment uitgevoerd in de vorm van een data mining wedstrijd, waarbij de deelnemers zo goed mogelijk moeten proberen te voorspellen en beschrijven wie er mogelijk interesse heeft in een caravanverzekering.

De omstandigheden lijken zoveel mogelijk op een data mining project in de praktijk. Om goede resultaten te verkrijgen en triviale modellen te vermijden is het belangrijk de data mining aanpak goed te laten aansluiten bij de business doelstelling (scoring in plaats van classificatie, weinig positieve gevallen). De gebruikte data set bevat een zeer klein aantal sterke voorspellers en een groot aantal variabelen met weinig of geen verband met het te voorspellen gedrag. Om realistisch (fout) gedrag aan het licht te kunnen laten komen worden er geen eisen gesteld aan het volgen van een nette, wetenschappelijke methodologie, er is een substantiële prijs uitgeloofd voor de winnaars en resultaten moeten verkregen worden onder aanzienlijke tijds-

druk.

De resultaten van de inzenders variëren inderdaad aanzienlijk. Een aantal inzendingen scoren niet veel beter dan random selectie, de beste inzenders identificeren bijna drie keer zoveel polisbezitters, iets meer dan de helft van de maximaal haalbare score. De meeste inzenders hebben ook een rapport ingeleverd over de gevolgde aanpak. Om de variëteit in resultaten in perspectief te plaatsen en dieper in te kunnen gaan op eventuele oorzaken van verschillen, is gekozen voor het raamwerk van bias variantie analyse.

De bias component van de fout hangt samen met de beperkingen van sommige methoden om bepaalde verbanden te kunnen representeren of vinden; een lineair model kan bijvoorbeeld niet goed niet lineaire verbanden uitdrukken. De variantiefout meet in welke mate methoden verschillende uitkomsten geven op willekeurige steekproeven, een probleem dat typisch veroorzaakt wordt door het feit dat er slecht beperkte data beschikbaar is.

Variantie blijkt een belangrijke component van de fout te zijn. Sommige deelnemers gebruiken strategieën in data preparatie, modellering en evaluatie om de variantie fout te minimaliseren, zoals variabele selectie, het gebruiken van simpele lage variantie methoden zoals naive Bayes en evaluatiemethoden zoals kruisvalidatie. Het toevoegen van variabelen in combinatie met gebrekkige variabele selectie, modelleren met complexe lage bias, hoge variantie methoden en uitvoerige experimentatie en finetuning met behoud van de 'beste' resultaten kunnen de variantie fout juist verhogen.

Het volgende hoofdstuk, hoofdstuk 5, betreft ook de evaluatiestap in het data mining proces. In dit hoofdstuk wordt een lans gebroken voor het ontwikkelen van methoden voor het evalueren en profileren van nieuwe data mining algoritmen. De introductie van nieuwe voorspelalgoritmen gaat soms gepaard met sterke claims over de voorspellende kracht. Het zogenaamde No Free Lunch theorema daarentegen stelt dat het niet mogelijk is dat een algoritme alle andere algoritmen verslaat voor alle soorten problemen. Een algoritme kan natuurlijk wel consistent slechter zijn, of beter op een bepaald subtype van problemen.

Een basis benchmark evaluatie bestaat vaak uit het vergelijken een nieuw algoritme met een aantal andere algoritmen over een random selectie van data sets. Dit is een nuttige, noodzakelijke test, maar uit het No Free Lunch theorema volgt dat dit hoogstens bewijs kan opleveren dat een nieuwe methode niet consistent slechter presteert, kortom dit is een minimale test. Om echt geaccepteerd te worden als een nieuw standaard gereedschap voor data mining toepassingen stellen we dat het belangrijk is om het gedrag van een methode te karakteriseren middels empirische testen, en dit te toetsen aan theoretische verwachtingen.

Als case voorbeeld wordt AIRS onderzocht, een zogenaamd Artificial Immune System of Immunocomputing algoritme. Dit algoritme is grofweg gebaseerd op een aantal principes die verklaren hoe het natuurlijk immuun systeem van hogere

organismen leert nieuwe indringers te herkennen, en ook een geheugen aanlegt van eerdere aanvallen. Om deze reden wordt het immuunsysteem ook wel het 'tweede brein' genoemd. We hebben voor deze klasse van methoden gekozen omdat bij nieuwe biologisch geïnspireerde algoritmen de nadruk soms meer op de biologische oorsprong ligt, dan op de kwaliteit van de modellen. Voor biologisch modelleren is dit natuurlijk geen probleem, wel voor praktijktoepassingen waarbij uiteindelijk alleen de prestatie telt in vergelijking met bestaande algoritmen.

In de papers die de in de eerste paar jaar over AIRS verschenen zijn, worden sterke claims gemaakt over de superieure prestatie van AIRS, een degelijk uitgevoerde benchmark ontbreekt echter. Als eerste stap is AIRS dan ook vergeleken met een reeks andere algoritmen over een aantal data sets. De uitkomst van deze experimenten is dat AIRS niet consistent beter is dan andere algoritmen zoals geclaimd, maar ook niet significant slechter. Dit is geen negatief resultaat, juist positief. Het geeft aan dat AIRS in principe robuust genoeg is om opgenomen te worden in het standaardassortiment van voorspellingsalgoritmen. Hierbij kan wel de vraag gesteld worden of de complexiteit van het algoritme in vergelijking met eenvoudigere nabuuralgoritmen gerechtvaardigd wordt door de resultaten.

Op de vraag wanneer AIRS een goed kandidaat algoritme is geeft de benchmark test geen antwoord. Interessanter wordt het dus als er manieren gebruikt worden om AIRS te karakteriseren en te profileren, en middels eenvoudige voorbeelden wordt aangetoond dat het niet moeilijk is hier specifieke methoden voor in te zetten en te ontwikkelen.

Een basis vraag is voor wat voor soort problemen AIRS een slechtere c.q. betere performance levert. Als voorbeeld wordt er een zogenaamde learning curve analyse uitgevoerd, waarin de hoeveelheid data die ter beschikking staat van een algoritme in stappen toeneemt, voor een gegeven voorspellingsprobleem. AIRS blijkt een vrij standaard curve te volgen, ietwat vlakker dan de gerelateerde zogenaamde nabuuralgoritmen. Dit bevestigt verwachtingen uit de theoretische vergelijking, gegeven dat AIRS abstraheert naar een kleiner aantal nabuur prototypen in plaats van alle training items als naburen te gebruiken.

De tweede vraag die beantwoord wordt, is op welke algoritmen AIRS het meeste en het minste lijkt in termen van het patroon van slechtere c.q. betere performance over verschillende data sets. Drie vergelijkingsmethoden worden gepresenteerd die allen een consistent beeld laten zien. Het gedrag van AIRS lijkt zoals theoretisch te verwachten is op het gedrag van nabuurmethoden.

Zoals aangegeven, AIRS is in dit geval slechts gebruikt als voorbeeld. De bedoeling is om modelprofilering als interessant en vrijwel braakliggend terrein van onderzoek onder de aandacht te brengen.

Concluderend, in deze dissertatie zijn een substantieel aantal specifieke deelonderwerpen onderzocht, maar al deze onderwerpen sluiten aan bij een klein aantal consistente kernboodschappen. Ten eerste wordt het belang benadrukt van het

onderzoeken van het start tot finish data mining proces, in plaats van enkel te concentreren op het ontwikkelen van nieuwe modelleeralgoritmen. De andere stappen in het proces zijn ook belangrijk, en het is mogelijk onderzoeksvragen te identificeren en methoden en technieken te ontwikkelen die een enkele praktijktoepassing overstijgen. Data fusie, modeldiagnose en -profilering zijn voorbeelden van zulke methoden. Een start tot finish visie, en het ontwikkelen van methoden voor alle fasen in het data mining proces zal het mogelijk maken belangrijke stappen voorwaarts te boeken, zoals data mining procesautomatisering, het aanbieden van data mining aan eindgebruikers in plaats van data mining experts en het inbedden van data mining in processen en systemen voor het nemen van automatische beslissingen. Dit zullen belangrijke factoren zijn in het verder laten opschalen van de praktijktoepassingen van data mining.

Chapter 8

Curriculum Vitae

Peter van der Putten was born in 1971 in Eindhoven, the Netherlands. He completed a masters degree in Cognitive Artificial Intelligence (Cognitieve Kunstmatige Intelligentie –'CKI') at Utrecht University in 1996 with a thesis on Self Organizing Map neural networks. Since then he has been working on applications of data mining and artificial intelligence for a number of companies including Sentient Machine Research, Frictionless Commerce, KiQ and currently for Chordiant Software. Since 1998 he has carried out part time academic research in the area of data mining at the Leiden Institute of Advanced Computer Science (LIACS) at Leiden University, first as a 'bursaal' and later as a guest researcher. In 2000 he also spent two months as a visiting researcher at the business school of MIT, the MIT Sloan School of Management.