DEPARTAMENT OF COMPUTER SCIENCE AND
ARTIFICIAL INTELLIGENCE
Univeristy of Granada


# Models of Supervised Classification. Applications to Genomics and Information Retrieval

by

**Andrés Ramón Masegosa Arredondo**

Dissertation submitted to the Department of Computer Science and Artificial Intelligence of the University of Granada as partial fulfilment of the requirements for the European PhD degree in Computer Science

Granada, May 2009

Editor: Editorial de la Universidad de Granada
Autor: Andrés Ramón Masegosa Arredondo
D.L.: GR 2312-2009
ISBN: 978-84-692-3115-9

UNIVERSITY OF GRANADA

Departament of Computer Science and Artificial Intelligence

# PhD Program: Design, Analysis and Applications of Intelligent Systems

PhD Thesis Dissertation:

Models of Supervised Classification. Applications to Genomics and Information Retrieval

PhD Student:

**Andrés R. Masegosa Arredondo**

Advisors:

**Andrés Cano Utrera, Serafín Moral Callejón**

A mis padres
(To my parents)

# Agradecimientos

Quiero aprovechar estas páginas para reconocer el apoyo y la dedicación de todas aquellas personas que, de alguna u otra manera, me han ayudado a la realización de esta tesis. Aunque también espero que este no haya sido el único momento, a lo largo de estos cinco años, en el que hayan recibido mi más sincero agradecimiento.

Por supuesto, he de comenzar agradeciendo la labor de mis dos tutores Andrés Cano y Serafín Moral por haber intentado constantemente transmitirme las enseñanzas, los conocimientos y la pasión necesarias para completar esta maravillosa aventura que es la realización de un doctorado. Les agradezco su paciencia, su esfuerzo y su buen hacer. Durante todo este tiempo me he sentido muy afortunado de haber trabajado con personas de su nivel profesional y su calidad personal.

Les agradezco también muy sinceramente a todos los miembros pertenecientes al grupo de investigación "Tratamiento de la Incertidumbre en Inteligencia Artificial" y los proyectos "Elvira", "Algra" y "Programo", el apoyo y el ejemplo de trabajo que me han sabido transmitir de muy diversas maneras y, especialmente, el haberme hecho sentir desde el primer día un miembro más del grupo. Me gustaría destacar especialmente a Manuel Gómez y a Joaquín Abellán con los que he compartido muy buenos momentos y de los que he recibido un excelente ejemplo de dedicación, trabajo y alegría. Agradezco sinceramente a Joaquín Abellán las enseñanzas recibidas durante las numerosas colaboraciones que hemos llevado a cabo. Todos esto me ha sido de una inestimable ayuda.

Continuo los agradecimientos mencionando a los doctores Joemon Jose y Hideo Joho de la Universidad de Glasgow, quienes me dieron un excelente trato durante todo el tiempo que duró mi estancia de investigación en esta universidad. A pesar de los pocos meses que allí pasé, ha sido una de las mejores experiencias de mi doctorado. Les agradezco sinceramente la dedicación y la atención que me ofrecieron y todas las valiosísimas enseñanzas que de ellos recibí. Una parte muy importante de esta tesis proviene de los diversos trabajos que he realizado con ellos.

Desde un punto vista personal, han sido muchas las personas, amigos, compañeros de trabajo y familiares, que emocionalmente me han apoyado. Si bien soportando mis momentos de dudas y decaimiento, también compartiendo muchos y muy buenos momentos de alegría y excelente compañía, que es realmente lo que me ha permitido realizar el gran esfuerzo y dedicación que un doctorado requiere. Quiero destacar a mi hermano Antonio, mis hermanas Luisa y Loly y, muy especialmente, a mis padres. Una vida no es suficiente para agradecerles todo lo que me han dado. Y quiero acabar agradeciendo, de una manera también muy especial, a mi novia, Antonia, todo el cariño y el apoyo incondicional que me ha brindado cada día durante todo este tiempo.

# Acknowledgements

I would like to take the opportunity to acknowledge the support and dedication to all those people who, in one way or another, helped me during my thesis. Although, over the last five years, too, I have often had reasons to be truly grateful to them.

Of course, I should begin by thanking my tutors Andrés Cano and Serafín Moral for their work and for constantly transferring to me the necessary teachings, knowledge and passion to finish this wonderful adventure which has culminated in my doctoral thesis. I am grateful for their patience, effort and hard work. All this time, I have felt fortunate to have worked with people of their professional and personal caliber.

I would also like to sincerely thank the people belonging to the research group "Uncertainty Treatment in Artificial Intelligence" and the projects "Elvira", "Algra" and "Programo", for the support and example they transmitted to me in many different ways and, especially, for making me feel like another member of the group from the very start. I am especially indebted to Manuel Gómez and to Joaquín Abellán with whom I have spent some great moments and from whom I have received an excellent example of dedication, work and joy. I sincerely wish to thank Joaquín Abellán for his teaching during our numerous collaborations. It has all been of inestimable help.

Thanks should also be given to doctors Joemon Jose and Hideo Joho of Glasgow University, who treated me so well throughout my research stay at this university. In spite of being there for just a few months, it was one of the best experiences of my doctorate. I wish to thank them sincerely for their dedication and attention and for all their valuable teachings. A very important part of this thesis is the result of the different work I did with them.

From a personal point of view, I have received emotional support from many people, friends, colleagues from the department and family members. With them I shared moments of doubt and dejection, as well as much joy and good company, which is really what gave me the strength required to write a doctoral thesis. Special thanks to my brother Antonio, my sisters Luisa and Loly and, in particular,

to my parents. A lifetime is insufficient to thank them for all they have given me. I would like to finish by giving my heartfelt thanks to my girlfriend, Antonia, for all the care and unconditional support she gave me every single day.

# Contents

# List of Figures

# List of Tables

# Part I

# Introduction

# Chapter 1

# Introduction

Supervised classification is an important field of data mining and machine learning research. It offers a wide range of different approaches to the problem of predicting the class of an object based on some indirect description of this object (i.e., automatically deciding whether an email is spam or not, by analyzing the words it contains).

Like many the developments in machine learning and data mining, Advances in supervised classification have made the leap from scientific papers in journals to real world applications of information processing and analysis. Nowadays, the spectrum of applications in which supervised classification models play a fundamental role is highly significant and includes very different areas such as spam email detection or analysis of gene expression data.

The range of available supervised classification models is actually huge. Each one has its strong and weak points depending on the characteristics demanded from them. Classification performance is one the most required characteristics along with good computational efficiency. In this respect, the naive Bayes classifier [54] and decision trees [27; 148] are very well known for their good trade-off between classification performance and time and memory efficiency. To the contrary, ensembles of models such as AdaBoost [64], Bagging [23] or AODE [176] stand out for their high performance but, in general, they involve higher computational resources.

Another relevant point is their interpretability or the information they provide on the intrinsic characteristics of the classification problem. This is a key

characteristic, because in many applications the most important objective involves extracting relevant knowledge from the classification problem (i.e. which genes are correlated with the survival rate of a cancer treatment). For example, those that perform a feature selection could provide information regarding which object's attributes are the most useful in the prediction task. Likewise, those based on Bayesian network models such as the Tree Augmented Network [68] or Sahami's K-Dependence classifier [155] offer very useful information on interdependence relationships among predictive attributes. One of the main reasons for the good reputation of decision trees is that they can be interpreted easily and naturally. To the contrary, those based on ensembles of models hardly provide any interpretable information, as they mainly act as a black-box.

Looking at the specific characteristics of the classification problem, we can also find many different situations in which the performance of classifiers can be affected. For example, Genomic problems usually involve a high number of predictive attributes or genes, along with a low number of samples from which the model must be inferred. Text classification problems also involve a high number of predictive features (i.e. words) but, in this case, the number of samples can be extremely high if we are dealing with data collections from the World Wide Web. This is what *No Free Lunch Theorem* [186] means: no classification model can claim to be the best one for all possible classification problems. Thus, when a new classification problem is faced, it is necessary in many cases to make some specific modification to the basic scheme in order to adapt the model to the particularities of the problem and to achieve a good classification performance.

The contributions of this thesis are based upon the above mentioned fact, i.e., that there is room to propose new classification models or to improve or adapt previously established ones on observing the particularities of specific supervised classification issues. This thesis proposes a set of new supervised classification models that attempt to behave optimally under the particular conditions in which they are applied.

## 1.1 Contributions of the Dissertation

### 1.1.1 Methodological Advances in Supervised Classification

Two contributions are presented in this part of the dissertation: a semi-Naive Bayes classifier that attempts to present high memory efficiency in the codification of the conditional probability distribution; and, secondly, we expose a Bayesian account of the problem of inferring classification trees.

Semi-Naive Bayes classifiers represent a wide and well studied family of probabilistic approaches to this problem. Research in this area has focused on improving the classification accuracy of these models. However, when some of these models are intended to be integrated in real software systems, other aspects aside from their classification performance should be considered. In Chapter 3, a new semi-Naive Bayes classifier is proposed whose main aim is to maintain a competitive classification performance while demanding few memory resources. We claim that this model is particularly suited for integration into the software of memory-limited devices.

Single classification trees and ensembles of classification trees represent another important family of models of supervised classification. In Chapter 4, we provide a Bayesian account of the problem of inferring this kind of models from a given dataset and address two particular issues related with these. Concretely, the problem of achieving well calibrated probability estimates of the class distribution in single classification trees is dealt with in Section 4.3. In an attempt to solve this problem, we employ a Bayesian approach that weights different rules of the induced tree in order to simulate a post-pruning process. Section 4.4 is devoted to ensembles of classification trees. In particular, we evaluated a random Bayesian split operator to build ensemble of trees. This method is inspired by one of the state-of-the-art tree ensemble models, Random Forest [26]. The aim of this new approach is to overcome the dependency of the random forest classifier on a particular parameter.

### 1.1.2 Applications to Genomics and Information Retrieval

This part of the dissertation starts with the application of supervised classification techniques to Genomics. Concretely, the problem faced here is the classification of gene expression data extracted from tumoral tissues affected by *diffuse large B-Cell lymphoma*. The particularities of this problem arise from the high number of predictive variables or genes involved, around eight thousand, along with the low number of samples, around two hundred. The problem consists of making accurate predictions and of showing which genes are more relevant for this task and can contain some biological information, and this must be achieved despite the high dimensionality and the low number of samples of this kind of data. Two supervised classification models especially designed for these issues are presented and evaluated.

The second application is related to the information retrieval field. Most applications of supervised classification techniques in this field involve document text classification problems and, recently, document ranking problems. However, this application focuses on a different area of information retrieval: the so-called *information retrieval in context*. More concretely, this part of the dissertation studies the role played by the so-called *contexts* in information retrieval tasks. Supervised classification models are employed in the core of a proposed methodology to measure the effect and dependency of contextual factors in contextual relevance modelling. Once again, the particularities of these data sets obliged us to adapt and design specific approaches for addressing this classification problem.

## 1.2 Overview of the Dissertation

The dissertation is arranged into five parts. The first, Part I, is an introduction containing two chapters. The first chapter, Chapter 1, provides an introduction to the dissertation detailing its main contributions. The second chapter, Chapter 2, is fully devoted to the definition and a bibliographical revision of supervised classification models. The basic notation used throughout the dissertation is also detailed here.

The second part, Part II, presents the two methodological contributions of this thesis. The memory efficient semi-Naive Bayes is presented in Chapter 3 while the Bayesian account of classification trees is depicted in Chapter 4.

The third part, Part III, contains the two chapters dedicated to the applications of classifiers to Genomics problems. The first chapter, Chapter 5, gives an introduction to gene expression data and the second one, Chapter 6, details the proposed approaches for the *diffuse large B-Cell lymphoma* classification problem.

Part IV focuses on information retrieval applications. Once again, two chapters are included in this part. One of these, Chapter 7, provides an introduction to *information retrieval in context* and Chapter 8 presents the methodology proposed for measuring the effect and dependency of contextual factors in relevance modelling.

Finally, Part V contains the last chapter, in which the main conclusions of the dissertation are discussed, as well as future research and publications supporting the contributions of this thesis.

# Chapter 2

# Supervised Probabilistic Classification

## 2.1 Introduction

The supervised classification problem can roughly be expressed as the prediction of a target feature of an object, given another set of features of the same object. In the probabilistic approach, all features are considered as random variables and their probability distribution must be inferred from a limited sample, the training set, and then the goodness of this estimate tested in a different sample set, the test set.

A random variable is a function that associates a numerical value with every outcome of a random experiment. Let $C$ denote the random variable to be classified or to be predicted with $k_c$ mutually exclusive and exhaustive classes $\{c_1, ..., c_k\}$; this variable will also be known as the class. Let $\mathbf{X} = (X_1, ..., X_n)$ represent an n-dimensional random variable in which each $X_i$ with $i = 1, ..., n$ is an unidimensional random variable. $\mathbf{X}$ corresponds to the set of features of the classification problem. Each feature is associated with one variable. An unlabeled instance of $\mathbf{X}$ is represented by $\mathbf{x} = (x_1, ..., x_n)$ and $(c, \mathbf{x})$ represents a labelled instance, that is to say, it is an assignation of $\mathbf{X}$ and the associated class of which has been established. In general, we use upper-case letters to denote random variables, lower-case letters to denote the values of the random variables and boldface letters to represent a vector of random variables or instances.

Finally, let $D = \{\vec{c}, \vec{x}\}$ denote a dataset with $T$ labelled instances $(c_j, \mathbf{x}_j)$ with $j = 1, ..., T$. $D$ will be the dataset from where the classification model will be inferred and evaluated. We also denote the mean number of values per attribute in $X$ as $\upsilon$.

## 2.2 Semi-Naive Bayes Classifiers

Bayesian classifiers [54] are well known and studied probabilistic classification methods. They predict the class label for an unlabelled instance $\mathbf{x} = (x_1, ..., x_n)$ using the probability of the variable $C$ conditioned to the feature vector $\mathbf{X}$. This probability is estimated by applying Bayes's Theorem as follows:

$$P(C|X_1, ..., X_n) = \frac{P(C)P(X_1, ..., X_n|C)}{\sum_{\mathbf{Y}} P(\mathbf{Y}|C)P(C)} \propto P(C)P(X_1, ..., X_n|C) \tag{2.1}$$

The class label with the maximum *a posteriori* probability is considered as the predicted class: $argmax_{c_i} P(x_1, ..., x_n|c_i)P(c_i)$. However, an accurate estimation of $P(X_1, ..., X_n|C)$ is by no means an easy task. This distribution has an exponential number of parameters in the number of variables and the number of samples $T$ is usually much lower. In order to deal with this problem, assumptions regarding independencies among the variables $X_i$ have to be made. Naive Bayes [54] is a simple approach assuming conditional independence among variables given the class $C$. Often, this assumption is unrealistic and violated by the classification features. In an attempt to relax this assumption and to tackle the problem of estimating the large conditional probability distribution, a new family of models known as semi-Naive Bayesian classifiers has arisen, each one presenting strong and weak points. Throughout the next section, the best known models of this wide family are presented. But first, by way of an introduction, the classic Naive Bayes classifier is detailed.

### 2.2.1 Naive Bayes

Naive Bayes (NB) [54; 118] is the simplest and best known approach. It assumes that all variables are conditionally independent from the class variable $C$. Hence, NB estimates the probability as follows:

$$P(C|X_1, ..., X_n) \propto P(C) \prod_{i=1}^{n} P(X_i|C)$$

In spite of this strict and unrealistic assumption, it usually exhibits very competitive performance. Some attempts have been made to explain this. Domingos [50] found some optimality conditions for the NB. The bias-variance decomposition of the error [187] also throws more light on the subject, claiming that the competitive error of NB is mainly due to their low variance component [195].

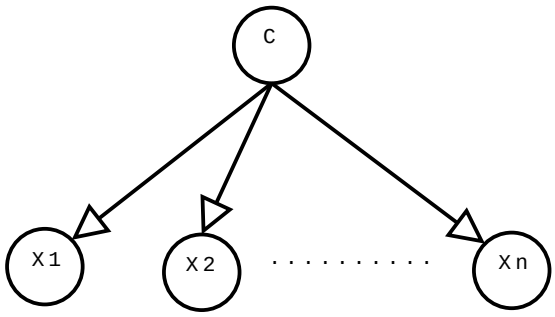


Figure 2.1: Naive Bayes

The following are the basic complexity measures:

- Training Time Complexity: $O(nT)$ (as the time needed to build the model).
- Classification Time Complexity: $O(nk_c)$ (as the time needed to classify a unlabelled instance).
- Training Space Complexity: $O(nvk_c)$ (as the memory space required to build the model).
- Model Space Complexity: $O(nvk_c)$ (as the memory space required to store the trained model).

### Gaussian Naive Bayes

When the domain of the data we work with is continuous and we do not want to lose information through discretization preprocessing, the variables involved must be considered as continuous random variables. In the Bayesian Networks field, this problem is tackled with the use of Gaussian Networks [160; 182]. In this kind of probabilistic graphical model, the joint density distribution of the continuous variables, given the discrete ones, is modelled as a multivariate Gaussian density. For the Naive Bayes model, this is seen in the assumption that the variables are normally distributed given the class variable. That is to say, the *a posteriori* probability of the class is evaluated with the following equation:

$$P(C = c_j|\mathbf{x}) \propto p(c_j) \cdot \prod_{i=1}^{n} f(x_i|C = c_j) \propto p(c_j) \cdot \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^i_{c_j}}} \cdot exp(-\frac{(x_i - \mu^i_{c_j})^2}{2\sigma^i_{c_j}})$$

where $\mu^i_{c_j}$ and $\sigma^i_{c_j}$ are the mean and the deviation values, respectively, of $X_i$ when $C = c_j$.

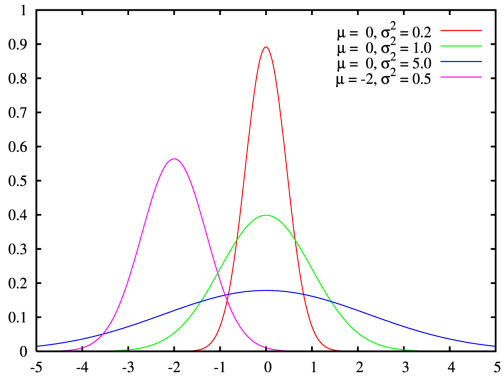


Figure 2.2: Density Functions of Normal Distribution

This model has been broadly analyzed and applied; see the following references for a complete review: [42; 51; 99; 130].

### 2.2.2 Selective Naive Bayes

NB uses the whole set of attributes to make predictions. But when two attributes are strongly correlated, NB overweights their particular predictions and, at the same time, reduces the influence of the other ones, which can cause bias in predictions. Removing the correlated and irrelevant attributes may improve the performance of the NB model.

Langley et al. [119] proposed two heuristic search methods for selecting an optimal subset of attributes where the goodness of that subset was evaluated with the leave-one-out cross validation error of the NB with these attributes:

**Forward Sequential Selection (FSS):** It starts with an empty set of attributes, and so the Bayesian classifier without attributes simply classifies all samples into the most frequent class in the data. Subsequently, each attribute not used in the current classifier is evaluated as a new attribute with a leaving-one-out cross validation. The best one, in terms of classification accuracy, is selected and the process is iteratively repeated until no improvement remained in the error with the inclusion of any of the available attributes.

**Backward Sequential Elimination (BSE):** This approach starts with a Bayesian classifier with all attributes. Each attribute used is then evaluated and the one with the best performance is removed. Once again, the process is iteratively repeated until there is no improvement in the error with the removal of any of the attributes.

Denoting $\mathbf{X}_s = \{X_{s_1}, ..., X_{s_p}\}$ to the final subset of selected features, the prediction is made assuming conditional independence among these attributes given the class variable:

$$P(C|X_1, ..., X_n) \propto P(C) \prod_{i=1}^{p} P(X_{s_i}|C)$$

The space and time complexity measures are the following [195]:

- Training Time Complexity: $O(n^2 T k_c)$

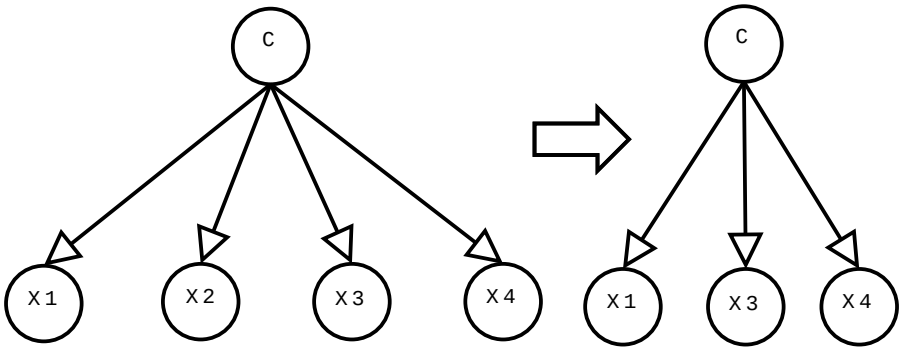


Figure 2.3: Selective Naive Bayes

- Classification Time Complexity: $O(nk_c)$
- Training Space Complexity: $O(nT + nvk_c)$
- Model Space Complexity: $O(nvk_c)$

### 2.2.3 Pazzani's semi-Naive Bayes

One way to deal with two dependent attributes is to remove one of them as in the previous model. Another approach would be to create compound attributes. Pazzani's semi-Naive Bayes [137] is based on a joining operation applied to attributes that attempts to relax the naive Bayes independence assumption creating new compound variables as a Cartesian product of two attributes. Assuming we have a classification problem where $\mathbf{X} = \{X_1, X_2, X_3\}$ and the class variable is $C$. If the independence assumption given the class is not maintained for $X_1$ and $X_3$, a better approximation for the Equation 2.1 shall be:

$$P(C = c|X_1 = x_1, X_2 = x_2, X_3 = x_3) \propto$$

$$\propto P(C = c)P(X_2 = x_2|C = c)P(X_1 = x_1, X_3 = x_3|C = c)$$

where $(X_1 \times X_3)$ could be considered as a single variable that maintains the assumption of independence in relation to $X_2$ given $C$. That is to say, this

approximation can be seen as a naive Bayes approach where some attributes are built as Cartesian products of those variables that are dependent given the class. Graphically, it can be seen that in Figure 2.4.

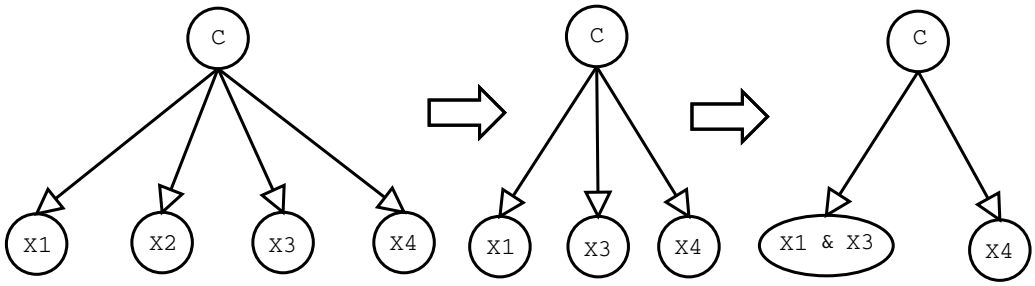


Figure 2.4: Pazzani's semi-Naive Bayes

Formally, if the resulting Cartesian product attribute set is denoted as $\{Joined_{g_1}, ..., Joined_{g_h}\}$, while the rest of original attributes that have been neither deleted nor joined are denoted as $\{X_{s_1}, ..., X_{s_p}\}$, the probability distribution is estimated as follows:

$$P(C|X_1, ..., X_n) \propto P(C) \prod_{i=1}^{g_h} P(Joined_{g_i}|C) \prod_{j=1}^{s_p} P(X_{s_j}|C)$$

The key point in this approach involves deciding which attributes should be joined, because a joining operation causes an increment in the number of parameters of the model and, in consequence, deterioration in the estimates of these parameters, along with greater risk of over-fitting. Pazzani [137] solves this problem by estimating the accuracy improvement caused by a joining operation. A joining operation is worthwhile when there is an estimated accuracy improvement for the Bayesian classifier with this new compound variable. The main disadvantage of Pazzani's approach is that the accuracy estimate is carried out using a leave-one-out cross validation on the training data, which implies a high computational cost.

Two search algorithms for model selection are evaluated by Pazzani:

**Forward Sequential Selection and Joining (FSSJ):** This starts with an empty set of attributes, and so the Bayesian classifier without attributes simply

classifies all samples to the most frequent class in the data. Subsequently, two operations are evaluated to obtain a new classifier:

**Independent Adding:** An attribute not used in the current classifier is evaluated as a new attribute conditionally independent of all other attributes used in the classifier.

**Forward Joining:** An attribute not used in the current classifier is evaluated to be joined with each attribute currently used in the classifier.

**Backward Sequential Elimination and Joining (BSEJ):** This approach starts with a Bayesian classifier with all attributes considered as conditionally independent. Two operations are then used to produce the next classifier:

**Deleting:** One attribute used in the current classifier is evaluated for removing.

**Backward Joining:** A pair of attributes used in the current classifier is replaced by a new compound attribute as joining of this pair.

In both cases, at each step, every possible operation is evaluated using a leave-one-out estimation on the training data set. If no change makes an improvement in the estimated accuracy, the current classifier is returned; otherwise, the most promising change is retained and the whole process is repeated.

The experimental study of [137] shows a slight improvement in the BSEJ scheme in relation to the FSSJ scheme, but with a huge computation cost which prohibits the latter methodology in many classification problems. The FSSJ scheme shows the best trade-off in terms of accuracy and efficiency.

In a subsequent study, Domingos et al. [50] attempted to show that an entropy-based metric for measuring the degree of dependence of the attributes is not a good criterion for joining variables. The measure used was the one considered by [112]:

$$D(X_i, X_j|C) = H(X_i|C) + H(X_j|C) - H(X_i, X_j|C)$$

where $H$ stands for the entropy with probabilities estimated from the relative frequencies in the learning sample $D$.

$$H(X_j|C) = \sum_c P(C=c) \sum_{x_j} -P(X_j = x_j|C=c) \cdot \ln P(X_j = x_j|C=c)$$

and analogously for the entropies $H(X_i|C)$ and $H((X_i, X_j)|C)$.

$D(X_i, X_j|C)$ measure is zero when $X_i$ and $X_j$ are completely independent given C and increases with their degree of dependence.

In an empirical study, they show that the semi-Naive Bayes method of [50] using this entropy-based measure for the joining criterion, rather than the estimated accuracy, does not significantly outperform the naive Bayes classifier in any of eleven UCI datasets. They finally suggest that accuracy-based metrics are better scores for joining variables than metrics that measure the degree of dependence between attributes.

Nonetheless, the main problem of Pazzani's approach [137] is the high-cost associated with accuracy-based metrics, as a cross validation process is performed at each step.

The basis complexity measures of FSSJ are the following [195]:

- Training Time Complexity: $O(Tn^3k_c)$
- Classification Time Complexity: $O(nk_c)$
- Training Space Complexity: $O(Tn + v^n k_c)$
- Model Space Complexity: $O(v^n k_c)$

### 2.2.4 Tree Augmented Naive Bayes (TAN)

Friedman et al. [68] attempted to build unrestricted Bayesian Networks as classification models. But when they compared them with NB they did not find any improvement in accuracy, and occasionally observed a deterioration. They argued that the huge space of models in which the classifier was sought may have been responsible for these poor results. They therefore opted for an intermediate

solution restricting the space model, by considering that each attribute could only depend, at most, upon one non-class attribute leading to a tree-like network or, as they called it, Tree Augmented Naive Bayes. They used conditional mutual information to find a maximum spanning tree [37] constituting the structure of the classifier.

If the parent of each attribute $X_i$ is indicated as $\pi(X_i)$, the probability distribution is estimated as:

$$P(C|X_1, ..., X_n) \propto P(C) \prod_{i=1}^{n} P(X_i|\pi(X_i), C)$$

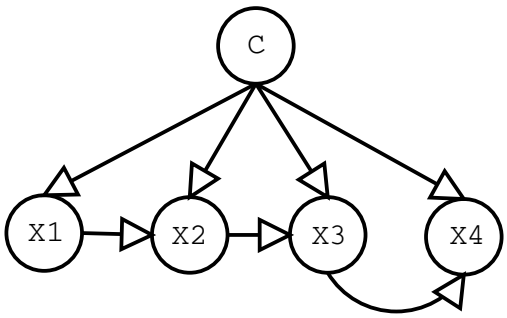


Figure 2.5: Tree Augmented Naive Bayes

Their basis complexity measures are the following ones [195]:

- Training Time Complexity: $O(n^2T + n^2v^2k_c + n^2logn)$
- Classification Time Complexity: $O(nk_c)$
- Training Space Complexity: $O(n^2v^2k_c)$
- Model Space Complexity: $O(nv^2k_c)$

### 2.2.5 K-Dependence Bayesian Classifier

Sahami [155] introduced a new kind of Bayesian classifier that attempts to generalize TAN, but not as generally as unrestricted Bayesian networks. Thus, he

introduces the concept of k-dependent classifiers as Bayesian networks-based classifiers where each attribute depends upon the class variable and, at most, on $k$ attribute variables. Therefore, NB can be considered as a 0-dependent classifier, while TAN would be a 1-dependent classifier. Thus, varying $k$, we can move from simpler to more complex classifiers. Sahami [155] also provided an algorithm to infer such models.

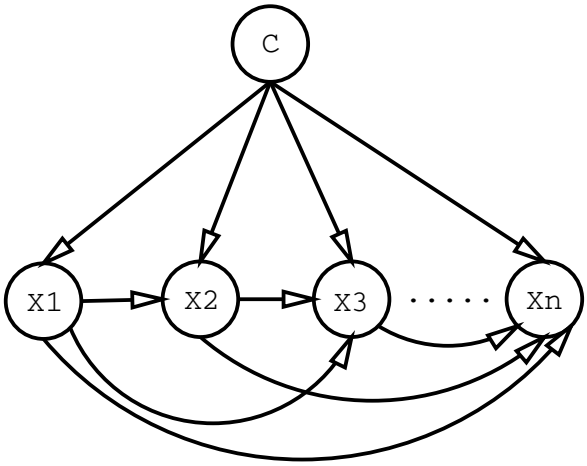


Figure 2.6: K-Dependence Bayesian Classifier

If the set of parents of each attribute $X_i$ is indicated as $\pi_k(X_i)$, the probability distribution is estimated as follows:

$$P(C|X_1, ..., X_n) \propto P(C) \prod_{i=1}^{n} P(X_i|\pi_k(X_i), C)$$

Their basis complexity measures are the following ([195]):

- Training Time Complexity: $O(n^2 T k_c v^2 + n(T + v^k))$
- Classification Time Complexity: $O(nk_c)$
- Training Space Complexity: $O((nv)^{k+1}k_c)$
- Model Space Complexity: $O(nv^{k+1}k_c)$

### 2.2.6 Average over One-Dependence Estimators (AODE)

AODE [176] appeared as a new approach intended to avoid problems associated with the selection of a specific model, as had occurred with previous approaches, while retaining the efficiency of one-dependence classifiers such as TAN. This classifier averages the predictions of all one-dependence classifiers that can be built with the attribute variables. At each one-dependence classifier, all attributes depend on the class and a single fixed attribute. Thus, by changing the fixed attribute that acts as parent of the remaining attributes, we obtain the complete set of one-dependence models. The estimated probability is performed as follows:

$$P(C|X_1, ..., X_n) \propto \sum_{j=1}^{n} P(C)P(X_j|C) \prod_{i=1}^{n} P(X_i|X_j, C)$$

It is considered as one of the state-of-the-art semi-Naive Bayes Classifiers, and outperforms most of the models presented in this section [195].

Their basis complexity measures are listed below:

- Training Time Complexity: $O(Tn^2)$
- Classification Time Complexity: $O(n^2k_c)$
- Training Space Complexity: $O(n^2v^2k_c)$
- Model Space Complexity: $O(n^2k_c)$

## 2.3 Decision Trees and Ensembles

### 2.3.1 Decision Trees

Decision Trees (also known as Classification Trees or hierarchical classifiers) have their origin in the work of [85], although it was after the publication of Quinlan's ID3 in 1979 [147] when they started to play an important role in machine learning. Subsequently, Quinlan also presented the algorithm C4.5 [148], which is an advanced version of ID3. Since then, C4.5 has been considered as a standard model in supervised classification. They have also been widely applied as a data analysis tool to very different fields, such as astronomy [156], biology [163], medicine [57; 60; 103; 111; 113; 128; 183; 188], physics [22], etc.

Decision trees are models based on a recursive partition method, the aim of which is to divide the data set using a single variable at each level. This variable is selected with a given criterion. They ideally come to define set of cases in which all the cases belong to the same class.

Their knowledge representation has a simple tree structure. It can be interpreted as a compact rule set in which each node of the tree is labelled with an attribute variable that produces a ramification for each one of its values. The leaf nodes are labelled with a class label, as can be seen in Figure 5.7.

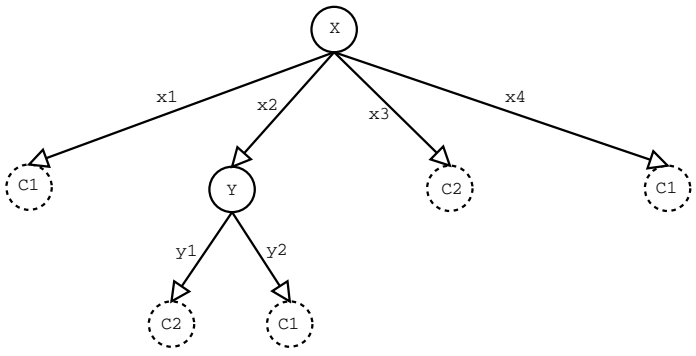


Figure 2.7: Decision Tree

The process for inferring a decision tree is mainly determined by the followings points:

- The criteria used for selecting the attribute to be placed in a node and ramified.
- The criteria for stopping the ramification of the tree.
- The method for assigning a class label or a probability distribution at the leaf nodes.
- The post-pruning process used to simplify the tree structure.

Many different approaches for inferring decision trees, which depend upon the aforementioned factors, have been published. Quinlan's ID3 ([150]) and C4.5 ([149]), along with the CART approach of [27], stand out among all of these.

One of the most important factors for the successful construction of a decision tree is the criterion used to select the split attribute at each node of the tree. Herein, we described the three best known and most used ones:

### Info Gain

This metric was introduced by Quinlan as the basis for his ID3 model [150] and is based on Shannon's entropy ([162]). This split criterion can therefore be defined for a variable $X$ given the class variable $C$ in the following way:

$$IG(X, C) = H(C) - H(C|X)$$

where $H(C)$ is the entropy of $C$: $H(C) = -\sum_j P(C = c_j) \log P(C = c_j)$, with $P(C = c_j)$ being the probability of each value of the class variable estimated in the training dataset restricted to the cases compatible with the path from the root to the actual node. In the same way, $H(C|X) = -\sum_x P(X = x) \sum_j P(C = c_j|x) \log P(C = c_j|x)$. Finally, we can obtain the following reduced expression for the Info-Gain criterion:

$$IG(X, C) = \sum_x \sum_j P(C = c_j, x) \log \frac{P(c_j, x)}{P(c_j)P(x)}$$

This criterion is also known as the *Mutual Information Criterion* and is widely used for measuring the dependence degree between an attribute variable and the class variable. Its weak point is that tends to select attribute variables with many states and consequently results in excessive ramification.

### Info Gain Ratio

In order to improve the ID3 model, Quinlan introduces the C4.5 model, in which the Info Gain split criterion is replaced by an Info Gain Ratio criterion which penalizes variables with many states. A procedure is also defined for working with continuous variables; it is possible to work with missing data, and a subsequent pruning process is introduced.

The Info-Gain Ratio of an attribute variable $X_i$ for a class variable $C$ can be expressed as:

$$IGR(X_i, C) = \frac{IG(X_i, C)}{H(X_i)}.$$

### Gini Index

This criterion is widely used in statistics for measuring the impurity degree of a partition of a dataset in relation to a given class variable (a partition can be said to be pure when it has only one single associated value of the class variable). The work of Breiman et al. [27] can be mentioned as a reference for the use of the Gini Index in decision trees.

In a given database, the Gini Index of a variable $X$ can be defined as:

$$gini(X) = \sum_x (1 - P(X = x)^2).$$

Thus, the split criterion based on the Gini Index is defined as follows:

$$GIx(X, C) = gini(C) - gini(C|X),$$

where

$$gini(C|X) = \sum_x P(X = x) gini(C|x)$$

### 2.3.2 Ensembles of Decision Trees

In many areas of science, such as Medicine or Finances, a second opinion, sometimes even more, is often sought before a decision is taken. Thus, we finally weight the individual opinions and combine them to take a final decision that should initially be more robust and trusted. This process of consulting "several experts" before making a decision has been also employed by the computational intelligence community. This approach is known by different names such as multiple classifier systems, committee of classifiers, mixture of experts or ensemble of classifiers, and has shown to produce much better results in comparison to single classifiers.

In this field, ensembles of decision trees appear to present the best trade-off among performance, simplicity and theoretic bases. The basic idea consists of generating a set of different decision trees and combining them with a majority vote criteria. That is to say, when an unlabelled unclassified instance arises, each single decision tree makes a prediction and the instance is assigned to the class value with the highest number of votes. In this way, a diversity issue appears as a critical point when an ensemble is built [24; 25]. If all decision trees are quite similar, the ensemble performance will not be much better than a single decision tree. However, if the ensemble is made up of a broad set of different decisions and exhibits good individual performance, the ensemble will become more robust, with a better prediction capacity [72].

There are many different approaches to this problem but Bagging [23], Random Forests [26] and AdaBoost [64] stand out as the best known and most competitive.

#### Bagging

Breiman's Bagging (bootstrap aggregating) [23] is one of the first cases of an ensemble of decision trees. It is also the most intuitive and simple and performs very well. Diversity in Bagging is obtained by using bootstrapped replicas of the original training set: different training datasets are randomly drawn with replacement. And, subsequently, a single decision tree is built with each training data replica with the use of the standard approach [27]. Thus, each tree can be

defined by a different set of variables, nodes and leaves. Finally, their predictions are combined by a majority vote. In Algorithm 1 a pseudocode description of this method is depicted.

**Algorithm 1** *Bagging Algorithm*

***Input:***

- *Training data $D$ with correct labels $c_1, ..., c_k$ representing the $k_c$ classes.*
- *A Decision Tree learning algorithm* ***LearnDecisionTree****,*
- *Integer* ***M*** *specifying number of iterations.*

***Do*** *m =1, ... ,M*

1. *Take a bootstrapped replica $D_m$ by randomly drawing from $D$.*
2. *Call* ***LearnDecisionTree*** *with $D_m$ and receive a single decision tree $DT_m$.*
3. *Add $DT_m$ to the ensemble,* ***E****.*

***End***

***Test: Simple Majority Voting*** *Given an unlabeled instance* ***x***

1. *Evaluate the ensemble $E = DT_1, ..., DT_M$ on $x$.*
2. *Let $v_{m,i} = 1$ if $DT_m$ predicts class $c_i$, $v_{m,i} = 0$ otherwise ($v_{m,i}$ be the vote given to class $c_i$ by classifier $DT_m$).*
3. *Obtain total vote received by each class*

$$V_i = \sum_{m=1}^{M} v_{m,i}, i = 1, ..., k_c$$

4. *Choose the class that receives the highest total vote as the final classification.*

### Random Forests

The idea of randomized decision trees was first proposed two decades ago by Mingers [129], but it was since the introduction of the ensemble of classifiers that the combination of randomized decision trees arose as a very powerful approach to supervised classification models [23; 26; 48]. A clear example of randomized

decision trees is the use of random split node selection. For example, Diettrich et al. [48] built ensemble of trees in which at each node the split was randomly selected from among the K best splits attributes.

**Random Forests** [26] is a combination of the bagging approach with a random split node selection. In this method, each decision tree was again built over a bootstrapped replicate of the former training set. But, as opposed to Diettrich et al.'s approach [48], K nodes were first randomly preselected and the best one was finally selected.

Thus, the Random Forests algorithm is similar to Bagging 's (Algorithm 1), changing the **LearnDecisionTree** algorithm for one that employs random split node selection. Algorithm 2 shows the pseudocode of this $K$-Random split method.

**Algorithm 2** *$K$-Random Split Node Selection*

***Input***

- *The available attribute variables* $\vec{X} = \{X_1, ..., X_p\}$
- *The method to compute the quality of a split attribute,* ***ComputeSplitScores*** *(usually Info Gain score or Gini Index).*

*end = false;*

***while*** *(not end)*

$\{S_1, ..., S_K\}$ = *Random Selection*($\vec{X}$);

$S^*$ = $argmax_i$ ***ComputeSplitScore***($S_i$);

$\vec{X} = \vec{X} \setminus \{S_1, ..., S_K\}$;

*if* $ComputeSplitScore(S^*) \geq 0$ *then*

*end=true;*

*else if* $\vec{X} = \emptyset$

$S^*$*=null; //Stop branching.*

*end=true;*

*return* $S^*$*;*

Random Forests outperformed Bagging and Diettrich approaches [26]. An issue of Bayesian Forests is their sensitivity to the selection of the $K$ value [26; 72], although Breiman suggested a $K$ value close to the logarithm of the number of variables as a default choice.

### AdaBoost

In 1990, Schapire demonstrated that a weak learner, an algorithm that generates classifiers that merely perform better than random guessing, can be transformed into a strong learner that can correctly classify all but an arbitrarily small fraction of the instances [158].

Similar to bagging, boosting also creates an ensemble of classifiers by resampling the data, which are then combined by majority voting. However, in boosting, resampling is strategically focused to provide the most informative training data for each consecutive classifier.

In 1997, Freund and Schapire introduced AdaBoost [64], a more general version of the original boosting algorithm. Among its many variations, AdaBoost.M1 is more commonly used to handle multiclass problems.

Roughly speaking, AdaBoost generates a set of single classifiers and combines them using a weighted majority voting of the predictions made by each individual classifier. The classifiers are learnt using instances drawn from an iteratively updated distribution of the training data. This distribution update forces instances misclassified by the previous classifier to be more likely to be included in the training data of the next classifier. Hence, the generated training data are focused towards those instances that are difficult to classify.

Algorithm 3 is a pseudocode description of this algorithm. As can be seen, the algorithm maintains a weighted distribution $D_k$ for the training instances. The training data subsets $S_t$ are drawn using this distribution and employed as training datasets to generate classifier $\Lambda_k$. At the start, the distribution is initialized to be uniform. Training error $\epsilon_k$ of $\Lambda_k$ is the sum of the distribution weights of the samples missclassified by $\Lambda_k$. This error must be less than $1/2$ in order to guarantee convergence.

Distribution $D_k$ is updated following the updated distribution rule depicted in Algorithm 3, which reduces the weight of well-classified instances and maintains the weights of the missclassified ones. This allows the classification efforts to be biased towards the instances that are difficult to classify. Finally, a majority voting is used to make the class prediction. The prediction of each classifier $\Lambda_k$ is a weight that depends upon its associated error rate, $\epsilon_k$.

**Algorithm 3** *Algorithm AdaBoost.M1*

***Input**:*

- *Sequence of $T$ examples $D = \{(x_i, c_i)\}$, $i = 1, \ldots, T$ with labels $c_i \in C$;*
- *Weak learning algorithm **WeakLearn**;*
- *Integer $K$ specifying number of iterations.*

***Initialize** $D_1(i) = \frac{1}{T}$, $i = 1, \ldots, T$*

***Do** for $k$ = 1, 2, . . . , K:*

- *Select a training data subset $S_k$, drawn from the distribution $D_k$.*
- *Train **WeakLearn** with $S_k$ , receive classifier $\Lambda_k$.*
- *Calculate the error of $\Lambda_k$:*
  $\epsilon_k = \sum_{i:\Lambda_k(\boldsymbol{x}_i)\neq c_i} D_k(i)$.
- *If $\epsilon_k > 1/2$, **abort**.*
- *Set $\beta_k = \epsilon_k/(1 - \epsilon_k)$.*
- *Update distribution $D_k$:*
  - $D_{k+1}(i) = \frac{D_k(i)}{Z_k}\beta_k$ *if* $\Lambda_k(\boldsymbol{x}_i) = c_i$.
  - $D_{k+1}(i) = \frac{D_k(i)}{Z_k}$, *otherwise.*

  *where $Z_k = \sum_i D_k(i)$ is a normalization constant chosen so that $D_{k+1}$ becomes a proper distribution function.*

***Test – Weighted Majority Voting:** Given an unlabeled instance $x$,*

- *Obtain total vote received by each class:*
  $V_j = \sum_{k:\Lambda_k(\boldsymbol{x})=c_j} \frac{1}{\beta_t}, j = 1, \ldots, k_c$.
- *Choose the class that receives the highest total vote $V_j$ as the final classification.*

## 2.4 Feature Selection in Supervised Classification

The classic induction methods of Bayesian classifiers use the whole set of attributes for the prediction tasks. But it is well known that the inclusion of redundant and irrelevant variables usually deteriorates the performance of the classifiers. This problem is particularly significant when in the domain problem there are a great number of variables, most of which are noisy or irrelevant. This occurs, for example, in gene expression data problems, where thousands of genes are analyzed but only a few are relevant.

Furthermore, feature selection approaches are also intended to extract the relevant features for a classification problem. In domain problems in which the number of variables is very high, as in the case of gene expression data, it is important to reach a high classification percentage, and to indicate which variables are relevant for the classification task.

Thus, in this section we provide a brief introduction to the main methods used for feature selection: filter and wrapper methods.

### 2.4.1 Filter Methods

The Filter approximation establishes an indirect goodness measure for the variables. Usually its output consists of a ranking among the predictive variables using this goodness measure, which scores the prediction capacity of the variables. Once the ranking is known, the K most important variables are used by the classifier. What is important is that the goodness measure is indirect, that is to say, it does not have any relation with the concrete classification model in which it will be incorporated.

Usually, these filter methods are employed as a preprocessing step in domains in which there are a high number of variables, most of these being irrelevant. Document text classification problems are a good example of this phenomenon, most of the words have no relationship with the categories in which documents are intended to be classified, and their presence can introduce much noise into the classifier. Moreover, the huge number of variables, very often around tens

of thousands words, does not allow the use of more complex or computationally expensive methods for word selection. Therefore, filter methods that are usually light are a very suitable option with regard to reducing this high number of irrelevant words.

There are many indirect goodness measures for the filter methods, a few of which are depicted here:

**Info Gain** This criterion is also known as the *Mutual Information Criterion* and is widely used for measuring the dependence degree between an attribute variable and the class variable and was already introduced as a criterion for selecting a variable in a node of a classification tree.

$$IG(X, C) = -\sum_{x}\sum_{j} P(C = c_j, x) \log \frac{P(c_j, x)}{P(c_j)P(x)}$$

**Info-Gain Ratio** In order to improve the Info-Gain split criterion, the Info-Gain Ratio criterion attempts to penalize variables with many states and was also introduced for classification trees:

$$IGR(X_i, C) = \frac{IG(X, C)}{H(X)}.$$

**Correlation-based Feature Selection (CFS)** This score [77] attempts to remove irrelevant Attributes, as well as redundant ones, selecting a subset of attributes that individually correlate well with the class $C$ but present little intercorrelation. The correlation between two categorical attributes $X_i$ and $X_j$ is measured using the *symmetric uncertainty* which is estimated as follows:

$$SU(X_i, X_j) = 2\frac{H(X_i) + H(X_j) - H(X_i, X_j)}{H(X_i) + H(X_j)}$$

where $H$ is the previously defined entropy function and $SU(X_i, X_j) \in [0, 1]$. So, CFS estimates the goodness of a set of attributes $(X_1, ..., X_p)$ as follows:

$$CFS(X_1, ..., X_p) = \frac{\sum_{i=1}^{p} SU(X_i, C)}{\sqrt{\sum_{i=1}^{p} \sum_{j=1}^{p} SU(X_i, X_j)}}$$

This score is usually employed with a greedy search because, as opposed to the previous ones, it is capable of evaluating a set of attributes as opposed to just one single one.

### 2.4.2 Wrapper Methods

Wrapper methods were developed in order to use the classifiers own classification accuracy as a direct measure, which would lead a search process across the space of all possible feature subsets.

Kohavi et al. [109] state that, as the aim of feature selection methods is to maximize the classification accuracy, features must be selected considering the classification model that will use them to make predictions. Furthermore, a set of attributes should receive a global score rather than giving an individual score to each one of them (the goodness of an attribute depends on the other selected variables).

One of the first approaches to Wrapper selection methods was made by [137] developing the so-called *selective Naive Bayes* (previously presented in Section 2.2.2 as another semi-Naive Bayes approximation). In this approach, a greedy search process was employed. It begins with an empty set of variables, and successively adds the variable that maximizes the accuracy of the classification, considering the candidate variable with the already selected ones. This value is obtained by the application of a Naive Bayes classifier using a *leave-one-out cross-validation* (LOO) scheme [168]. In this validation method, the classifier is iteratively trained and tested with different training and data sets. In each iteration, one single instance is removed from the data set. The remaining dataset will be employed to train the classifier and this classifier is then tested with the extracted single instance. This process is repeated and one different instance in each iteration is removed: as many times as there are different instances in the

whole dataset. Finally, the average value is reported as the estimated classification accuracy. The search process stops when the addition of more variables does not suppose an increment in the classification accuracy. Thus, the variables selected in the iterated process are the most relevant variables selected by this approach.

Other authors employ similar techniques [109; 190], showing the great classification potential of these methods. But the main disadvantage of wrapper methods is their high computational cost, mainly due to the fact that the classification accuracy has to be estimated with a k-fold-cross validation method or, as in Pazzani [137], with an even more expensive leave-one out procedure. Wrapper methods therefore have to build and to evaluate a classifier model many times over, which results in a very costly approach.

Many attempts have been made to reduce the computational cost of these wrapper approaches [108; 132]. One of the most effective methods involved the application of a filter approach aimed at selecting a reduced feature set as a starting point for a wrapper approach.

# Part II

# Methodological Advances

# Chapter 3

# A Memory Efficient Semi-Naive Bayes Classifier with Grouping of Cases

In this chapter, we present a semi-Naive Bayes classifier that searches for dependent attributes using different filter approaches. In order to prevent the number of cases of the compound attributes from being excessively high, a grouping procedure is always applied after the merging of two variables. This method attempts to group two or more cases of the new variable into a single one, in order to reduce the cardinality of the compound variables. As a result, the presented model is a competitive classifier, particularly in terms of quality of class probability estimates, with very low memory requirements. Thus, we believe it would be an interesting candidate model for supervised classification systems integrated in limited-memory devices such as mobile phones.

## 3.1 Motivation

In recent times, a new trend in applications of supervised classification systems has been emerging in the field of mobile computing. The use of smart-phones, car boarded computers, portable computers, etc. is becoming more commonplace and, consequently, better and more sophisticated functionalities are demanded by users. Many of these new functionalities will be based on the data managed by

these devices. It is therefore easy to see how data mining techniques and, especially, supervised classification systems will be integrated into the core of their loaded software. Research in this field has discovered advanced functionalities such as speaker classification [59], multi-modal interaction [121], activity detection with a tri-axial accelerator [36], ubiquitous healthcare [127], etc., applications in which supervised classification techniques were applied to successfully achieve them.

For this new kind of applications, the computational resources demanded by classification systems are critical due to the fact that the computational and memory capacities of these devices are much less powerful than classical PC's. The restrictions in power consumption are also very important, and applications that demand intensive CPU utilization will therefore have to be discarded.

In the probabilistic approach to supervised classification, all predictive features are considered as random variables in the training process and the classification is made by the modelling of the probability distribution of the class variable conditioned to these predictive features (see Section 2.2 for details). In order for the estimation of this probability distribution to be feasible, several hypothesis must be assumed, and several approaches have been proposed to achieve a better approximation of the probability distribution. Hence, many different classification systems have been proposed in the literature (Section 2.2 gives an overview of some of these classifiers). The success or failure of each approach depends on the specific features of each classification problem (No Free Lunch Theorem [186]). However, some of these approaches stand out as very competitive classifiers in a wide range of classification problems.

Furthermore, maximization of the classifiers performance has constituted the main aim in this area of research, and many of the proposed approaches are associated with significant increments in computing and memory resources. Classifier ensembles are a clear example of performance maximization with higher computational costs (for details see Section 2.3.2).

In this sense, many studies involving performance evaluation, comparison, analysis, etc. of probabilistic classifiers have been published [33; 34; 195]. Naive Bayes stands out as a competitive classifier presenting high computational efficiency and, as has previously been stated, to the contrary, ensemble classifiers

appear to exhibit the highest success classification rates, but present an overload of computational resources [23; 64]. As an intermediate point, the large family of the so-called semi-Naive Bayes classifiers shows the best trade-off between computational efficiency and competitive performance. In particular, those based upon a mixture of models such as AODE [176], WOADE [97], HNB [193], LBR [196], etc. (for an extensive comparative study see [195]) or upon uncertainty measures founded on imprecise probabilities, such as maximum entropy [2].

But most of these studies do not focus on, or have not considered, the possibility of these classifiers being embedded in a mobile computer with strict memory-limited capacities. Moreover, although space complexity analysis under big O notation has been derived for many different predictors, to our knowledge no studies compare the number of parameters needed for each classifier in a wide range of classification problems. Big O notation gives information in the upper limit and this limit is sometimes far from the expected one. Decision trees are a clear example of this situation; their space complexity is O(t), t being the number of training examples (one leaf of the tree per sample in the worst case), while most of the times the post-pruning process dramatically reduces this number [149].

For applications running on memory-limited devices, there is therefore a need to review the criteria for establishing the most suitable approaches for classification systems.

The aim of this chapter is two-fold. Firstly, to evaluate the specific memory requirements of a representative range of probabilistic classifiers in order to provide insights with regard to their trade-off between performance and memory requirements. And, secondly, to introduce a new semi-Naive Bayes approach that exploits the grouping of the cases of the new compound variables in order to achieve highly competitive performance with a very low memory load. With the support of an intensive experimental evaluation, we claim that this approach represents an excellent model to be used in classification applications running on memory-limited devices.

The rest of the chapter is organized as follows. In Section 3.2 we introduce the proposed semi-Naive Bayes with grouping of cases, detailing the different metrics evaluated. Subsequently, in Section 3.3, we perform a memory space analysis of different classification models. These classifiers are then empirically compared

with our proposal in Section 3.4. We conclude by giving the main conclusions of this study and future research in Section 3.5.

## 3.2 A Semi-Naive Bayes Classifier with Grouping of Cases

### 3.2.1 An initial overview to the approach

Our approach is inspired by the ideas of Pazzani's semi-Naive Bayes (see Section 2.2.3 for details). Roughly speaking, Pazzani's approach joined variables with a Cartesian product when they were dependent and removed those that were not informative to the class variable.

One the main problems associated with the process of joining variables is that the number of possible cases increases exponentially with the number of merged variables. Another of the problems specifically associated with Pazzani's approach was the employment of accuracy-based measures to decide which variables should be joined, with the use of cross validation methods for computing these scores. Hence, although it constitutes a powerful classification approach, it becomes computationally prohibited for classification problems with a high number of variables.

Aiming to address these issues, we first employ efficient filter measures to decide the joining of two variables, along with a complementary grouping process to reduce the number of cases of these new compound variables. One of the main effects of this grouping of cases will be a significant reduction in the number of parameters that this classification model employs to encode the probability distribution. Let us examine a simple example of this idea.

**Example 1** *Let us suppose than random variables $X$ and $Y$ have two cases $\{x_0, x_1\}$ and $\{y_0, y_1\}$, respectively. If $X$ and $Y$ are not statistically independent, given the class variable $C$ (the independence assumption of Naive Bayes), this approach will join them in a new compound variable $X \times Y$ with the following cases $\{(x_0, y_0), (x_0, y_1), (x_1, y_0), (x_1, y_1)\}$. The grouping process will attempt to group those cases that present similar information. For example, Grouping$(X \times Y)$*

*could be a simpler variable with the following three cases* $\{\{(x_0, y_0) + (x_1, y_0)\}, (x_0, y_1), (x_1, y_1)\}$ *if* $(x_0, y_0)$ *and* $(x_1, y_0)$ *provided the same information on the class* $C$.

Three different filter measures (see Section 2.4.1 for an introduction to filter measures) were empirically evaluated to decide the joining of two variables or the groping of two cases. The aim was to select an optimal configuration of filter measures for the joining and the grouping processes.

The rest of the sections are organized as follows: Section 3.2.2 describes the joining process and Section 3.2.3 the grouping method. Finally, In Section 3.2.4 the experimental results are shown.

### 3.2.2 Joining criteria

Along the lines of Pazzani's or Kononenko's approaches [112; 137], all possible pairs of variables are considered at each step with a given metric. The metrics we propose evaluate the convenience of joining two variables with respect to keeping them separated. Thus, the most suitable ones are merged by creating a new compound variable with the Cartesian product of the value sets of the original ones. This procedure is used in an iterative fashion: the old joined variables are removed and the new one is included as a candidate to be joined once again with another variable. The process continues until there are no more variable pairs to verify the fixed joining criterion.

In this study, we propose three filter metrics as a **joining criterion**. Each one has a **joining condition** (**JC**$(X_i, X_j)$) that tests whether the variables $X_i$ and $X_j$ can be joined, along with a **joining metric** (**JM**) that selects the most suitable pair to be joined.

#### Bayesian Dirichlet equivalent Metric (BDe)

Bayesian scoring criteria have been widely used to choose from several alternative models [80], because of the inherent penalty that they impose on the more complex models in order to prevent against over-fitting.

Bayesian scores measure the quality of a model, $M$, as the posterior probability of the model providing the learning data $D$. The logarithm of this quantity is ususally considered for computational reasons, giving rise to:

$$Score(M : D) = \ln P(M|D) = \ln P(D|M) + \ln P(M) - \ln P(D)$$

This value can be computed under a suitable hypothesis. The BDe (Bayesian Dirichlet equivalent) [80] assumes a uniform prior probability over the possible models and a prior Dirichlet distribution over the parameters with independence for the parameters of different conditional distributions. A global sample size, $S$, is usually considered and we then assume that for each variable $Z$, the a priori probability of the vector $(P(z))_z$ is Dirichlet with the same parameters $S/k_Z$ for all values $P(z)$, where $k_Z$ is the number of possible values of $Z$.

The metric for joining attributes $X_i$ and $X_j$ is computed as the difference: $Score(M_1 : D) - Score(M_2 : D)$, where $M_1$ is a Naive Bayes model in which $X_i$ and $X_j$ are joined and $M_2$ a model in which they are considered conditionally independent given the class. Under global sample size $S$, this difference can be computed as:

$$\mathbf{JM}_{BDe}(X_i, X_j) = \sum_c \ln\left(\frac{\Gamma(S/k_C)}{\Gamma(S/k_C + N_c)}\right)\left(T_{C,X_i,X_j} - T_{C,X_i} - T_{C,X_j}\right)$$

where

$$\begin{aligned} T_{C,X_i,X_j} &= \sum_{x_i,x_j} ln\frac{\Gamma(S/(k_C k_{X_i} k_{X_j}) + N_{cx_i x_j})}{\Gamma(S/(k_C k_{X_i} k_{X_j}))} \\ T_{C,X_k} &= \sum_{x_k} ln\frac{\Gamma(S/(k_C k_{X_k}) + N_{cx_k})}{\Gamma(S/(k_C k_{X_k}))} \end{aligned}$$

$\Gamma(.)$ is the gamma function ($\Gamma(\alpha + 1) = \alpha.\Gamma(\alpha)$), $N_{cx_i x_j}$ is the number of occurrences of $(C = c, X_i = x_i, X_j = x_j)$ in the learning sample $D$ (analogously for $N_c$ and $N_{cx_k}$).

The pair $X_i, X_j$ with greatest metric is selected and the attributes are merged if the joining condition is verified i.e.

$$\mathbf{JC}_{BDe} = [JM_{BDe} > 0]$$

### The Expected Log-likelihood under Leaving-one-out (L10)

The score of a model $M_i$ for a dataset $D$ is obtained by adding for each vector of cases $(\mathbf{x}, c) \in D$, the logarithm of $P(c|\mathbf{x})$, where the probability $P$ is obtained by estimating the parameters of $M_i$ with $D - \{(\mathbf{x}, c)\}$. That is, an estimation of the log-likelihood of the class [136] is carried out with a wrapper leaving-one-out procedure.

The metric for joining attributes $X_i$ and $X_j$ should be computed as the difference of scores between the model in which $X_i$ and $X_j$ are joined and the model in which they are considered conditionally independent given the class. However, this value can depend on the remaining attributes and can be difficult to compute in a closed form. For this reason, we compute it considering that only variables $X_i$ and $X_j$ and $C$ are included in the model. This can be considered as an approximation which allows rapid computation. This metric is computed as:

$$\begin{aligned}\mathbf{JM}_{L1O}(X_i, X_j) &= \sum_{c,x_i,x_j} N_{cx_ix_j} \left[\ln\left(\frac{P^*(x_i, x_j|c)P^*(c)}{\sum_{c'} P^*(x_i, x_j|c')P^*(c')}\right)\right] - \\ &- N_{cx_ix_j} \left[\ln\left(\frac{P^*(x_i|c)P^*(x_j|c)P^*(c)}{\sum_{c'} P^*(x_i|c')P^*(x_j|c')P^*(c')}\right)\right]\end{aligned}$$

where the probabilities $P^*$ are estimated from the sample using the Laplace correction and discounting 1 in the absolute frequencies of values $(c, x_i, x_j)$ in the sample:

$$P^*(x_i, x_j|c) = \frac{N_{x_ix_jc}}{N_c + k_{X_i}k_{X_j} - 1}, \quad P^*(c) = \frac{N_c}{N + k_C - 1}, \quad P^*(x_k|c) = \frac{N_{x_k}}{N_c + k_{X_k} - 1}$$

and for $c' \neq c$:

$$P^*(x_i, x_j|c') = \frac{N_{x_i x_j c'} + 1}{N_{c'} + k_{X_i} k_{X_j}}, \quad P^*(c') = \frac{N_{c'} + 1}{N + k_C}, \quad P^*(x_k|c') = \frac{N_{x_k} + 1}{N_{c'} + k_{X_k}}$$

In this way, we assume that attributes $X_i$ and $X_j$ are suitable for joining if the following condition is met:

$$\mathbf{JC}_{L1O}(X_i, X_j) = [JM_{L1O}(X_i, X_j) > 0]$$

### Log-likelihood Ratio Test (LRT)

The last approach for deciding when to join two variables is based on a log-likelihood ratio test [184]. The log-likelihood ratio test (LRT) has been used to compare two nested models, $M_1$ and $M_2$ (in this case $M_1$ is the model with merged variables and $M_2$ the simpler model with conditionally independent variables). The log-likelihood ratio criterion is expressed by:

$$LRT = -2 \ln \left( \frac{\sup_\theta P_{M_2}(D|\theta)}{\sup_\theta P_{M_1}(D|\theta)} \right) = -2 \sum_{c,x_i,x_j} N_{cx_ix_j} \ln \left( \frac{N_{cx_i} N_{cx_j}}{N_c N_{cx_ix_j}} \right)$$

where $P_{M_i}(D|\theta)$ is the likelihood of the data under the model $M_i$ and the parameter $\theta$. The $\sup_\theta P_{M_i}(D|\theta)$ is obtained by computing the likelihood of the data when parameters are estimated with maximum likelihood (in this case, the parameters are equal to the relative frequencies in the sample). The third part of the equality shows the closed form for computing $LRT$.

$LRT$ is asymptotically distributed as a Chi-square random variable with a number of degrees of freedom equal to the difference in the number of parameters between the two models.

In this case, $LRT$ follows a chi-square distribution with $(k_{X_i} - 1)(k_{X_j} - 1)k_C$ degrees of freedom [184], where $k_{X_i}$ is the number of cases of $X_i$. The null hypothesis ($H_0$) of the test is that $X_i$ and $X_j$ are independent given the class. A significance level $\alpha$ is considered and this $LRT$ metric is computed as the p-value of the following test:

$$\mathbf{JM}_{LRT}(X_i, X_j) = \chi^2_{(k_{X_i}-1)(k_{X_j}-1)k_C}(LRT)$$

The associated criterion is that the null hypothesis is rejected. But the question is whether this test is valid for the comparison of two models, while in this algorithm it is applied many times to the $\frac{n(n-1)}{2}$ possible variable pairs ($n$ is the actual number of active variables), increasing the possibilities of an error of the LRT. In order to avoid this effect, the $\alpha$ level is divided by a corrector factor $\rho = \sum_{t=1..R} \frac{1}{t}$ that hinders the rejection of null hypothesis [175].

Thus, the joining criterion condition is expressed by:

$$\mathbf{JC}_{LRT}(X_i, X_j) = [JM_{LRT}(X_i, X_j) > (1 - \frac{\alpha}{\rho})]$$

where $R = \frac{n(n-1)}{2}$ is the number of tests, $\mathbf{JM}_{LRT}$ is the joining criterion metric and $\mathbf{JC}_{LRT}$ is the joining criterion condition.

### The Joining Algorithm (JA)

This algorithm corresponds to the process for joining dependent variables in a recursive form. It considers the three different joining criteria (i.e. BDe, L10, LRT). The process is quite simple: it joins the variables with the highest score when the joining condition is verified.

**Algorithm 4** *Joining Algorithm (**JA**)*

$\boldsymbol{Z} = \{X_1, ..., X_n\}$;

$end$ = *false*;

*while* ($|\mathbf{Z}| \geq 2 \wedge \neg end$)

$\{X_i, X_j\} = arg\max_{\{X_r, X_s\}}\{\boldsymbol{JM}(X_r, X_s) : X_r, X_s \in Z\}$;

*if* $\boldsymbol{JC}(X_i, X_j)$ *then*

$T = X_i \times X_j$;

$\boldsymbol{Z} = \boldsymbol{Z} \setminus \{X_i, X_j\}$;

$\boldsymbol{Z} = \boldsymbol{Z} \cup \{T\}$;

*else*

*end=true;*

*return* $\boldsymbol{Z}$*;*

### 3.2.3 Grouping process

As has already been stated, an important problem with regard to joining two attributes is that the number of necessary parameters is much greater, for example, if $X_i$ and $X_j$ are considered independent given $C$, $(k_{X_i} + k_{X_j} - 2)k_C$ parameters have to be estimated, while if $X_i$ and $X_j$ are joined $(k_{X_i}k_{X_j} - 1)k_C$ parameters should be estimated. For example, if we join two variables with 7 possible values and two classes, the resulting combined variable will need 96 values. If for some of these combinations there are very few samples in the learning data, the estimations of these parameters will not be very reliable.

To solve this problem, we propose a mechanism for grouping similar cases of an attribute. We apply it to each variable resulting from a joining operation and before any other joining of variables is considered. Thus, we attempt to reduce the number of cases before computing the Cartesian product with another variable, in order to avoid a combinatorial explosion in the number of cases and increasing the possibility of further combinations of this variable.

The process consists of evaluating each pair of cases using a given criterion (based on the same principles used in the joining process), and when the presence of two cases does not suppose a significant benefit with respect to considering them as a unique case, they will be grouped in a single case. The aim of this approach is obviously to reduce the complexity introduced into the model with the joining of the variables.

#### Grouping criteria

The same principles that we used in the joining process are valid with regard to defining new criteria for this purpose.

To fix the notation, let $X$ be the considered variable. We assume that $x_i$ and $x_j$ are two possible cases of this variable. $X_{(i,j)}$ will be the variable in which cases

$x_i$ and $x_j$ have been grouped into a single case. We only consider the subsample $D_{X(i,j)}$ given by samples of the original dataset $D$ in which it is verified that $X = x_i$ or $X = x_j$. In order to make the criterion independent from the other possible cases of variable $X$ and their frequencies, we suppose that $x_i$ and $x_j$ are the only possible values of variable $X$. In this situation, the grouping of $x_i$ and $x_j$ implies the definition of a new attribute with a single case. A variable with only one case is useless. Therefore, the grouping criteria will check whether $X$ (with two possible values $x_i, x_j$) is relevant to $C$ under sample $D_{X(i,j)}$. $M_1$ will be the more complex model with $X$ relevant to $C$ and $M_2$ will represent the simpler model in which $X$ is irrelevant to $C$ ($x_i$ and $x_j$ have been grouped).

The metric for grouping $x_i$ and $x_j$ into a single case is denoted by $\mathbf{GM}(x_i, x_j)$ and the condition by $\mathbf{GC}(x_i, x_j)$. Thus, the three grouping criteria we propose are:

**BDe score (BDe)** The difference between the BDe scores of making $X$ dependent or independent of $C$ produces:

$$\mathbf{GM}_{BDe}(x_i, x_j) = T_{x_iC} + T_{x_jC} - T_C$$

where

$$T_{x_kC} = \ln\left(\frac{\Gamma(\frac{S}{2})}{\Gamma(\frac{S}{2} + N_{x_k})}\right) + \sum_c \ln\left(\frac{\Gamma(\frac{S}{2k_C} + N_{cx_k})}{\Gamma(\frac{S}{2k_C})}\right)$$

$$T_C = \ln\left(\frac{\Gamma(S)}{\Gamma(S + N_{x_i} + N_{x_j})}\right) + \sum_c \ln\left(\frac{\Gamma(\frac{S}{k_C} + N_c)}{\Gamma(\frac{S}{k_C})}\right)$$

In these expressions $S$ is a parameter (global sample size) and the frequencies are measured in subsample $D_{X(i,j)}$.

Thus, we assume that the $x_i$ and $x_j$ are suitable for grouping (grouping condition) when:

$$\mathbf{GC}_{BDe}(x_i, x_j) = [GM_{BDe}(x_i, x_j) \leq 0]$$

**Leave one-out score (L10)** As before, we compute the logarithm of the likelihood of the learning sample $D_{X(i,j)}$ under the two models $M_1$ and $M_2$, but for each estimation of the likelihood of a case, we remove this case from the sample where the estimations of the rest of the parameters are carried out (this can be done by decreasing the associated frequencies by 1). We employ the Laplace correction to estimate these probabilities. Hence, the resulting formula is expressed by:

$$\begin{aligned}\mathbf{GM}_{L10}(x_i, x_j) &= \sum_c N_{cx_i} \left( \ln \frac{N_{cx_i}}{N_{x_i} + k_C - 1} \right) + \sum_c N_{cx_j} \ln \left( \frac{N_{cx_j}}{N_{x_j} + k_C - 1} \right) \\ &- \sum_c N_c \ln \left( \frac{N_c}{N_{x_i} + N_{x_j} + (k_C - 1) - 1} \right)\end{aligned}$$

And the grouping condition is expressed as in the BDe metric:

$$\mathbf{GC}_{L10}(x_i, x_j) = [GM_{L10}(x_i, x_j) \leq 0]$$

**Log-Likelihood Ratio Test (LRT)** As in Section 3.2.2, we apply the log-likelihood ratio test to compare models $M_1$ and $M_2$. The statistic is:

$$\begin{aligned}LRT &= -2 \sum_c N_{cx_i} \ln \left( \frac{N_{cx_i}(N_{x_i} + N_{x_j})}{N_c N_{x_i}} \right) \\ &\quad -2 \sum_c N_{cx_j} \ln \left( \frac{N_{cx_j}(N_{x_i} + N_{x_j})}{N_c N_{x_j}} \right)\end{aligned}$$

The null hypothesis ($H_0$) is the simpler model $M_2$ (the model where cases $x_i$ and $x_j$ are grouped). In this case $LRT$ follows a chi-square distribution with $(k_C - 1)$ degrees of freedom. The grouping metric is the p-value of this test:

$$\mathbf{GM}_{LRT}(x_i, x_j) = \chi^2_{k_C-1}(LRT)$$

The associated criterion is that the null hypothesis is rejected. Such as the equivalent joining criterion, the $\alpha$ level is divided by a corrector factor $\rho = \sum_{t=1...R} \frac{1}{t}$ where $R = \frac{k(k-1)}{2}$ is the number of tests and $k$ is the number of active cases in the variable $X$ [175].

$$\mathbf{GC}_{LRT}(x_i, x_j) = [GM_{LRT}(x_i, x_j) > (1 - \frac{\alpha}{\rho})]$$

### The grouping algorithm (GA)

This algorithm is the process for grouping the irrelevant cases of a variable in a recursive form. The only variation is the grouping criterion considered. The similarity to Algorithm 4 should be noted. In both cases, a model selection and a model transformation are carried out.

**Algorithm 5** *Grouping Algorithm (**GA**)*

$S_X = \{x_1, ..., x_n\}$;

*end = false;*

*while*($|\mathbf{S_X}| \geq 2 \wedge \neg end$)

$\{x_i, x_j\} = arg\max_{\{x_r,x_s\}}\{\boldsymbol{GM}(x_r, x_s) : \{x_r, x_s\} \in S_X\}$;

*if* $\boldsymbol{GC}(x_i, x_j)$ *then*

$t = \{x_i \cup x_j\}$;

$S_X = S_X \setminus \{x_i, x_j\}$;

$S_X = S_X \cup \{t\}$;

*else*

*end=true;*

*return* $S_X$;

In the experimental results, presented in Table 3.3 of Section 3.2.4, we will see that there are significant differences between the three proposed criteria. But the main advantage of the introduction of this grouping process is the reduced complexity of the final model as well as the time needed to build the classifier. The experimental results show a reduction of 50% in time.

## The classifier

Once we have described the joining and grouping processes, we depict how to compose these two processes:

**Algorithm 6** *Semi-Naive Bayes with Grouping of Cases (**Semi-NB-G**)*

$\boldsymbol{Z} = \{X_1, ..., X_n\}$;

$end = false$;

*while* $(|\mathbf{Z}| \geq 2 \wedge \neg end)$

$\{X_i, X_j\} = arg\max_{\{X_r, X_s\}}\{JM(X_r, X_s) : \{X_r, X_s\} \in Z\}$;

*if* $JC(X_i, X_j)$ *then*

$T = Grouping(X_i \times X_j)$;

$\boldsymbol{Z} = \boldsymbol{Z} \setminus \{X_i, X_j\}$;

$\boldsymbol{Z} = \boldsymbol{Z} \cup \{T\}$;

*else*

*end=true;*

*return Grouping(**Z**);*

As can be seen, the grouping process is applied each time that two attributes to be joined are selected. At the end, we apply the grouping method again to all attributes with the aim of processing the attributes that have not been selected for joining.

As three distinct metrics can be used in the joining and grouping process, there are nine possible schemes to define the classifier. In the next section, the experimental evaluation will show how the LRT criterion in the joining process and the L1O in the grouping process provide the best results.

As a final analysis, we will show that the combination of the joining and grouping procedures presents an important potential. It can perform some additional preprocessing tasks as side effects, as is shown in the following example.

**Example 2** *We performed the following experiment in Elvira environment [39]. We considered three binary variables,* $C, X, Y$*, where* $C$ *is the class variable and* $X$ *and* $Y$ *the classifying attributes. Let us assume that the possible cases of* $C$ *are* $\{c_0, c_1\}$*, the possible cases of* $X$ *are* $\{x_0, x_1\}$*, and the possible cases of* $Y$*:* $\{y_0, y_1\}$*. We built a Bayesian network in which* $C$ *and* $Y$ *are conditionally independent given* $X$*, more specifically, we considered the following graph structure:* $C \longrightarrow X \longrightarrow Y$*. The probability tables are given in such a way that there is a high degree of dependence between* $C$ *and* $X$ *and* $X$ *and* $Y$*. For example, assuming that* $P(x_i|c_i) = 0.8, \quad i = 0, 1$*,* $P(y_i|x_i) = 0.85, \quad i = 0, 1$*. Then, we obtained a random sample of size 1000 from the joint probability distribution by simulation, using logic sampling [82].*

*We subsequently applied the proposed combined joining-grouping procedure (any metric will work in this situation) to the random sample. The result is that first,* $X$ *and* $Y$ *are joined in only one variable, as they are not conditionally independent given the class. The new variable will take values in set* $\{(x_0, y_0), (x_0, y_1), (x_1, y_0), (x_1, y_1)\}$*. Then grouping is applied. The fact that* $C$ *and* $Y$ *are conditionally independent given* $X$*, means that* $(x_0, y_0), (x_0, y_1)$ *are grouped into one single value and* $(x_1, y_0), (x_1, y_1)$ *grouped into another value. The first is equivalent to* $X = x_0$ *and the second equivalent to* $X = x_1$*.*

*The final effect of the two steps is that, in order to build a Naive Bayes, variable* $Y$ *is discarded, and only* $X$ *is kept. That is, this approach is capable of removing irrelevant variable* $Y$ *(given* $X$*). Therefore, as can be seen in this example, an implicit procedure exists for eliminating variables in this approach.*

### 3.2.4 Experimental evaluation

In this section, we make an initial empirical evaluation to establish which joining and grouping metrics of the ones proposed here achieve the best trade-off between classification accuracy and quality of probabilities estimates. The reduction in the memory requirements proposed in the introduction of this chapter will be subsequently evaluated in Section 3.4.

Table 3.1: Data Bases

| Name | Cases | Attributes | Classes |
|---|---|---|---|
| anneal | 898 | 39 | 5 |
| balance-scale | 625 | 5 | 3 |
| german-credit | 1000 | 21 | 2 |
| diabetes | 768 | 9 | 2 |
| glass | 214 | 10 | 6 |
| heart-statlog | 270 | 14 | 2 |
| ionosphere | 351 | 35 | 2 |
| iris | 150 | 5 | 3 |
| lymphography | 148 | 19 | 4 |
| sonar | 208 | 61 | 2 |
| vehicle | 846 | 19 | 4 |
| vowel | 990 | 12 | 11 |
| zoo | 101 | 17 | 7 |

Accordingly, we used a reduced set of 13 UCI datasets detailed in Table 3.1. We did not take datasets with missing values because they were not considered in the development of the proposed metrics (more exactly, only missing values that were randomly distributed could be considered) and we attempted to prevent this problem from disturbing the conclusions. Furthermore, as we will see throughout this section, the small differences among many of the metrics led us to choose an evaluation methodology that allows close inspection, at a dataset level, of these differences.

For the experiments, we used Elvira environment [39] and Weka platform [185]. The continuous values were discretized by the Fayyad & Irani method [58] using Weka filters themselves. We employed two evaluations or performance measures in this experimental evaluation: the classical prediction accuracy (noted as %); and, to evaluate the precision of probability class estimates, we computed the logarithm of the likelihood of the true class as: *Log-Likelihood*$= \ln(P(\hat{c}_i|\mathbf{x}))$ (noted as $LL$).

The evaluation of the classifiers was achieved with a 10-fold-cross validation repeated 10 times scheme for each database. Thus, 100 train and test evaluations were obtained. With these estimates, we carried out the comparison among classifiers in each dataset using a corrected paired t-test [134] implemented in

Weka with a 5% statistically significant level. In this way, we fixed a classifier as a reference (marked with $\star$) and then each proposed classifier was compared against it. The result of the corrected paired t-test can show a statistically significant improvement or Win (denoted as Wins (**W**) or by the symbol $\circ$), a not statistically significant difference or Tie (denoted as Ties (**T**)) and a statistically significant deterioration (denoted as Defeats (**D**) or by the symbol $\bullet$) in the evaluation measures for each dataset.

A summary of the comparison was made by adding the times that the proposed classifier obtains a statistically significant difference with respect to the reference classifier in accordance with the corrected paired t-test for each dataset. These results are shown in the rows starting with *W/T/D*. (E.g., in Table 3.2 (a), $JA_{BDe}$ obtains a statistically significant improvement or win in the accuracy with respect to $NB$ in 2 databases (*anneal* and *vehicle*, noted as $\circ$). While $JA_{L10}$ obtains a statistically significant deterioration in the Log-Likelihood with respect to $NB$ in 4 databases (*german-credit, lymphography, vowel and zoo*, noted as $\bullet$)).

### Evaluating joining criteria

To this end, we compared the results provided by the Joining Algorithm (**JA**) using the three proposed criteria (without grouping) in respect to the performance of the Naive Bayes classifier, Table 3.2(a), and in respect to Pazzani's semi-Naive Bayes classifier, Table 3.2(b).

As is shown in Table 3.2(a), the BDe and LRT criteria outperform the Naive Bayes classifier in terms of accuracy in two datasets, whereas the LRT loses in the *balance-scale* dataset. Observing log-likelihood in the same table, BDE and LRT appear to perform better. Indeed, the joining process is designed to minimize the log-likelihood metric and achieves this reduction in a robust manner, and no significant deterioration is observed in any of the datasets for the two metrics.

To the contrary, when we compare with Pazzani's semi-Naive Bayes, Table 3.2(b), these two metrics only lose in terms of accuracy and log-likelihood in one dataset, whereas they beat this semi-Naive Bayes in terms of log-likelihood in four datasets. The LRT criterion appears to be slightly better than BDe in terms of log-likelihood although we do not find any sound reason to prefer any of them.

Table 3.2: Joining Criteria Evaluation

| Dataset | ⋆ NB | $J_{BDe}$ | $J_{L1O}$ | $J_{LRT}$ | ⋆ NB | $J_{BDe}$ | $J_{L1O}$ | $J_{LRT}$ |
|---|---|---|---|---|---|---|---|---|
| anneal | 95.95 | 97.82○ | 96.09 | 97.29 | -0.13 | -0.10 | -0.14 | -0.09○ |
| balance-scale | 71.56 | 71.48 | 69.40● | 69.40● | -0.60 | -0.60 | -0.54○ | -0.54○ |
| german-credit | 75.04 | 74.54 | 70.23● | 73.99 | -0.53 | -0.53 | -0.61● | -0.54 |
| pima-diabetes | 75.26 | 75.27 | 73.31 | 75.18 | -0.54 | -0.52○ | -0.53 | -0.51 |
| Glass | 71.94 | 70.32 | 67.35 | 71.50 | -0.91 | -0.90 | -1.03 | -0.88 |
| heart-statlog | 82.56 | 81.81 | 75.11● | 82.89 | -0.47 | -0.44○ | -0.49 | -0.40○ |
| ionosphere | 89.40 | 89.86 | 65.78● | 90.09 | -1.62 | -1.04○ | -0.50○ | -0.81○ |
| iris | 93.33 | 93.20 | 91.33 | 93.33 | -0.22 | -0.21 | -0.29 | -0.22 |
| lymphography | 85.10 | 85.75 | 59.00● | 86.17 | -0.43 | -0.40 | -0.82● | -0.38 |
| sonar | 76.71 | 75.46 | 55.82● | 74.98 | -0.84 | -0.60○ | -0.67 | -0.56○ |
| vehicle | 61.06 | 68.63○ | 43.31● | 68.88○ | -2.00 | -0.68○ | -1.21○ | -0.68○ |
| vowel | 61.99 | 63.22 | 62.36 | 66.57○ | -1.01 | -0.99 | -1.83● | -0.93○ |
| zoo | 93.98 | 92.29 | 91.39 | 93.88 | -0.12 | -0.16 | -0.39● | -0.12 |
| Average | 79.53 | 79.97 | 70.81 | 80.32 | -0.73 | -0.55 | -0.70 | -0.51 |
| W/T/D | | 2/11/0 | 0/6/7 | 2/10/1 | | 5/8/0 | 3/6/4 | 7/6/0 |

(%)Percent of cases corrected classified (LL) Mean Log-Likelihood

○, ● statistically significant improvement or degradation

(a) Naive Bayes comparison respect to Joining Algorithm (J) with the tree proposed joining criteria: BDe, L10 and LRT

| Dataset | ⋆ SemiNB | $J_{BDe}$ | $J_{L1O}$ | $J_{LRT}$ | ⋆ SemiNB | $J_{BDe}$ | $J_{L1O}$ | $J_{LRT}$ |
|---|---|---|---|---|---|---|---|---|
| anneal | 97.41 | 97.82 | 96.09 | 97.29 | -0.13 | -0.10 | -0.14 | -0.09 |
| balance-scale | 72.33 | 71.48 | 69.40● | 69.40● | -0.69 | -0.60○ | -0.54○ | -0.54○ |
| german-credit | 72.99 | 74.54 | 70.23● | 73.99 | -0.54 | -0.53 | -0.61● | -0.54 |
| pima-diabetes | 74.45 | 75.27 | 73.31 | 75.18 | -0.52 | -0.52 | -0.53 | -0.51 |
| Glass | 70.15 | 70.32 | 67.35 | 71.50 | -0.86 | -0.90 | -1.03● | -0.88 |
| heart-statlog | 79.63 | 81.81 | 75.11 | 82.89 | -0.47 | -0.44 | -0.49 | -0.40 |
| ionosphere | 90.15 | 89.86 | 65.78● | 90.09 | -0.33 | -1.04● | -0.50● | -0.81● |
| iris | 93.40 | 93.20 | 91.33 | 93.33 | -0.24 | -0.21 | -0.29 | -0.22 |
| lymphography | 79.69 | 85.75 | 59.00● | 86.17 | -0.58 | -0.40○ | -0.82● | -0.38○ |
| sonar | 71.39 | 75.46 | 55.82● | 74.98 | -0.56 | -0.60 | -0.67 | -0.56 |
| vehicle | 67.36 | 68.63 | 43.31● | 68.88 | -0.74 | -0.68 | -1.21● | -0.68 |
| vowel | 67.51 | 63.22● | 62.36● | 66.57 | -1.40 | -0.99○ | -1.83● | -0.93○ |
| zoo | 88.38 | 92.29 | 91.39 | 93.88 | -0.52 | -0.16○ | -0.39 | -0.12○ |
| Average | 78.83 | 79.97 | 70.81 | 80.32 | -0.58 | -0.55 | -0.70 | -0.51 |
| W/T/D | | 0/12/1 | 0/6/7 | 0/12/1 | | 4/8/1 | 1/6/6 | 4/8/1 |

(%)Percent of cases corrected classified (LL) Mean Log-Likelihood

(b) Semi-NB comparison respect to Joining Algorithm (J) with the tree proposed joining criteria: BDe, L10 and LRT

However, as this analysis shows, the L10 performance is clearly a bad one. We have found no sound reason for this. A possible one might be that L10 criterion tests the joining of two variables without considering the remaining ones. The experiments are performed with the full set of attributes and the final behaviour can therefore differ from what was initially expected.

### Evaluating grouping criteria

With the analysis of the previous subsection, we selected the LRT criterion as the most suitable one for joining variables. We therefore fixed it and performed the same process in order to evaluate which criterion was the most suitable for grouping cases using the semi-Naive Bayes with Grouping of Cases (**SNB-G**). In Table 3.3, we show the results with the three possible grouping criteria. It can be seen that there is almost no difference among them either. Perhaps the L1O criterion stands out a little more. Curiously, this metric did not present the same problem when it was applied to joining variables. Perhaps this is due to the fact that grouping is a simpler operation than joining.

Table 3.3: Grouping Criteria Evaluation.

| Dataset | $\star$ NB | $G_{BDe}$ | $G_{L1O}$ | $G_{LRT}$ | $\star$ NB | $G_{BDe}$ | $G_{L1O}$ | $G_{LRT}$ |
|---|---|---|---|---|---|---|---|---|
| anneal | 95.95 | 98.36◦ | 98.13◦ | 98.01◦ | -0.13 | -0.06◦ | -0.07◦ | -0.08◦ |
| balance-scale | 71.56 | 73.13 | 73.08 | 73.13 | -0.60 | -0.52◦ | -0.52◦ | -0.52◦ |
| german-credit | 75.04 | 74.29 | 74.58 | 75.09 | -0.53 | -0.54 | -0.54 | -0.53 |
| pima-diabetes | 75.26 | 73.71 | 74.61 | 74.06 | -0.54 | -0.52 | -0.51 | -0.51 |
| Glass | 71.94 | 70.98 | 72.01 | 70.38 | -0.91 | -0.93 | -0.89 | -0.93 |
| heart-statlog | 82.56 | 83.56 | 83.26 | 83.11 | -0.47 | -0.40◦ | -0.40◦ | -0.41 |
| ionosphere | 89.40 | 88.83 | 88.92 | 89.58 | -1.62 | -0.44◦ | -0.49◦ | -0.38◦ |
| iris | 93.33 | 93.33 | 93.33 | 93.40 | -0.22 | -0.22 | -0.22 | -0.21 |
| lymphography | 85.10 | 84.34 | 85.58 | 81.04 | -0.43 | -0.41 | -0.41 | -0.48 |
| sonar | 76.71 | 75.69 | 75.45 | 75.64 | -0.84 | -0.56◦ | -0.55◦ | -0.54◦ |
| vehicle | 61.06 | 68.15◦ | 69.19◦ | 66.86◦ | 2.00 | -0.71◦ | -0.70◦ | -0.76◦ |
| vowel | 61.99 | 63.37 | 67.07◦ | 61.39 | -1.01 | -0.98 | -0.90◦ | -1.02 |
| zoo | 93.98 | 95.65 | 92.71 | 95.07 | -0.12 | -0.11 | -0.13 | -0.12 |
| Average | 79.53 | 80.26 | 80.61 | 79.75 | -0.73 | -0.49 | -0.49 | -0.50 |
| W/T/D | | 2/11/0 | 3/9/0 | 2/11/0 | | 6/7/0 | 7/6/0 | 5/8/0 |

(%)Percent of cases corrected classified (LL) Mean Log-Likelihood

**Description:** NB comparison respect to SNB-G with the $J_{LRT}$ joining criterion and the three proposed Grouping Criteria: BDe, L10 and LRT.

Thus, we fixed the metric LRT as the joining metric and the metric L10 for the grouping process. Once again, we wish to point out that the small differences

among the metrics (except for the case of L10 in the joining method) indicates that other combinations might also be valid. We did not evaluate them in order not to introduce more complexity into this section.

### Effects of the grouping process

In this subsection we attempt to provide some results showing the effects of the introduction of the grouping process. Thus, we compare the approach with joining and grouping against the approach with only the joining of variables.

Table 3.4 shows the comparison in terms of classification accuracy and log-likelihood while Table 3.5 shows the effect of grouping in terms of model training and testing time required by both classifiers.

In this first analysis, introduction of the grouping process does not have any clear effect in terms of accuracy and log-likelihood. Although, in both cases, some improvement is noted. It is in relation to training time where we can see the best effect of the grouping process. The time needed to train a model is reduced on average by half, while for some datasets, this reduction is greater and a deterioration is never observed.

Table 3.4: Performance Improvement by Grouping Introduction

| Dataset | ⋆ Join. Alg. | **SNB-G** | | ⋆ Join. Alg. | **SNB-G** | |
|---|---|---|---|---|---|---|
| anneal | 97.29 | 98.13 | | -0.09 | -0.07 | |
| balance-scale | 69.40 | 73.08 | ∘ | -0.54 | -0.52 | ∘ |
| german-credit | 73.99 | 74.58 | | -0.54 | -0.54 | |
| pima-diabetes | 75.18 | 74.61 | | -0.51 | -0.51 | |
| Glass | 71.50 | 72.01 | | -0.88 | -0.89 | |
| heart-statlog | 82.89 | 83.26 | | -0.40 | -0.40 | |
| ionosphere | 90.09 | 88.92 | | -0.81 | -0.49 | ∘ |
| iris | 93.33 | 93.33 | | -0.22 | -0.22 | |
| lymphography | 86.17 | 85.58 | | -0.38 | -0.41 | |
| sonar | 74.98 | 75.45 | | -0.56 | -0.55 | |
| vehicle | 68.88 | 69.19 | | -0.68 | -0.70 | |
| vowel | 66.57 | 67.07 | | -0.93 | -0.90 | |
| zoo | 93.88 | 92.71 | | -0.12 | -0.13 | |
| Average | 80.32 | 80.61 | | -0.51 | -0.49 | |

(%) Percentage of cases corrected classified (LL) Mean Log-likel.

∘, • statistically significant improvement or degradation

Table 3.5: Time Reduction by Grouping Introduction

| Dataset | Join. Alg. | **SNB-G** | Join. Alg. | **SNB-G** |
|---|---|---|---|---|
| anneal | 4.83 | 3.12 | 0.35 | 0.23 |
| balance-scale | 0.03 | 0.02 | 0.01 | 0.01 |
| german-credit | 1.00 | 0.58 | 0.14 | 0.08 |
| pima-diabetes | 0.11 | 0.07 | 0.02 | 0.02 |
| Glass | 0.06 | 0.04 | 0.01 | 0.00 |
| heart-statlog | 0.08 | 0.06 | 0.01 | 0.01 |
| ionosphere | 2.97 | 1.27 | 0.22 | 0.15 |
| iris | 0.01 | 0.01 | 0.00 | 0.00 |
| lymphography | 0.19 | 0.13 | 0.01 | 0.01 |
| sonar | 0.41 | 0.29 | 0.04 | 0.03 |
| vehicle | 4.06 | 0.92 | 0.42 | 0.14 |
| vowel | 0.78 | 0.49 | 0.07 | 0.05 |
| zoo | 0.08 | 0.06 | 0.00 | 0.00 |
| Average | 1.12 | 0.54 | 0.10 | 0.06 |
| (T) Model Training (seconds) | | | (T) Model Testing | |

### Comparing with Pazzani semi-Naive Bayes

Finally, we compare our proposed semi-Naive Bayes with Pazzani's approach. We attempt to show that our approach is a competitive one, although it does not use wrapper measures capable of measuring the global effect of the joining of two variables.

The results in terms of accuracy and log-likelihood are shown in Table 3.6 and the model training time is shown in Table 3.7.

Table 3.6: Pazzani's Semi-NB, **SNB-G** and NB Comparison

| Dataset | ⋆ Semi-NB | **SNB-G** | NB | ⋆ Semi-NB | **SNB-G** | NB |
|---|---|---|---|---|---|---|
| anneal | 97.41 | 98.13 | 95.95 | -0.13 | -0.07 ○ | -0.13 |
| balance-scale | 72.33 | 73.08 | 71.56 | -0.69 | -0.52 ○ | -0.60 ○ |
| german-credit | 72.99 | 74.58 | 75.04 | -0.54 | -0.54 | -0.53 |
| pima-diabetes | 74.45 | 74.61 | 75.26 | -0.52 | -0.51 | -0.54 |
| Glass | 70.15 | 72.01 | 71.94 | -0.86 | -0.89 | -0.91 |
| heart-statlog | 79.63 | 83.26 | 82.56 | -0.47 | -0.40 | -0.47 |
| ionosphere | 90.15 | 88.92 | 89.40 | -0.33 | -0.49 ● | -1.62 ● |
| iris | 93.40 | 93.33 | 93.33 | -0.24 | -0.22 | -0.22 |
| lymphography | 79.69 | 85.58 ○ | 85.10 ○ | -0.58 | -0.41 ○ | -0.43 ○ |
| sonar | 71.39 | 75.45 | 76.71 | -0.56 | -0.55 | -0.84 ● |
| vehicle | 67.36 | 69.19 | 61.06 ● | -0.74 | -0.70 | 2.00 ● |
| vowel | 67.51 | 67.07 | 61.99 ● | -1.40 | -0.90 ○ | -1.01 ○ |
| zoo | 88.38 | 92.71 | 93.98 | -0.52 | -0.13 ○ | -0.12 ○ |
| Average | 78.83 | 80.61 | 79.53 | -0.58 | -0.49 | -0.73 |
| W/T/D | | 1/12/0 | 1/10/2 | | 5/7/1 | 4/8/3 |
| (%)Percent of cases corrected classified | | | | (LL) Mean Log-Likelihood | | |

○, ● statistically significant improvement or degradation

Table 3.7: Semi-Naive Bayes Comparison - Model Training Time (seconds)

| Dataset | Semi-NB | **SNB-G** |
|---|---|---|
| anneal | 370.07 s | 3.12 s |
| balance-scale | 0.31 s | 0.02 s |
| german-credit | 26.34 s | 0.58 s |
| pima-diabetes | 1.16 s | 0.07 s |
| Glass | 0.97 s | 0.04 s |
| heart-statlog | 0.85 s | 0.06 s |
| ionosphere | 32.34 s | 1.27 s |
| iris | 0.05 s | 0.01 s |
| lymphography | 5.15 s | 0.13 s |
| sonar | 8.73 s | 0.29 s |
| vehicle | 48.27 s | 0.92 s |
| vowel | 28.24 s | 0.49 s |
| zoo | 1.96 s | 0.06 s |
| Average | 40.34 s | 0.54 s |

In Table 3.6, we can see that our proposed semi-Naive Bayes performs in a similar way to Panzzani's semi-Naive Bayes in terms of percentage of correct classifications. Our approach performs better in terms of log-likelihood in 5 datasets and only loses in one dataset. Furthermore, it can also be observed that our algorithm is more robust than Pazzani's classifier when compared to the Naive Bayes classifier.

The other import aspect to be pointed out is the computational cost of these two approaches. In Table 3.7 we show the training time of the two approaches in each data set. As can be seen, there is a drastic time reduction in some databases with a high number of attributes: *anneal* with a reduction of 99.2% of training time, *german-credit* with a reduction of 97.8%, *vehicle* with a reduction of 98.1%, etc. And the average model training time reduction in relation to Pazzani's semi-Naive Bayes is of 98.7%.

### Conclusions of the experimental evaluation

In this section, we have presented a combination of two procedures as a preprocessing step for a Naive Bayes classifier: a method for joining variables and a method for grouping cases of the new compound variables.

We have proposed and evaluated three different metrics as joining and grouping metrics. The first one was the very well known log-likelihood ratio test, where a new corrector factor for its significant level was introduced. The second one was the well-studied Bayesian Dirichlet equivalent metric. And, finally, we introduced a new proposal, which has been called expected log-likelihood under leaving-one-out (the estimation of the log-likelihood of the model was carried out with a wrapper leaving-one-out procedure via a derived closed form). Under our experimental analysis, the first one was the most suitable for joining variables and the last one the most suitable for grouping cases.

We have also shown that the combined application of a joining and grouping process obtains a similar performance to similar wrapper methods in terms of accuracy and better in terms of log-likelihood. But the main advantage of this model is the great model training time reduction, particularly in datasets with a higher number of variables (see Table 3.1 and Table 3.7): we achieve an average reduction of 98.7% of training time, keeping the simplicity of a semi-Naive Bayes approach based on filter measures.

Throughout the following sections, we will show how, apart from its competitive performance, this classification model is capable of encoding the underlying class probability distributions using a very low number of parameters.

## 3.3 Memory Space Analysis of Classification Models

In this section, we expound a memory space analysis focusing on the classification stage. We derive a theoretical analysis based on the big O notation as well as the exact number of parameters of the final learnt classifier when this computation is feasible. We performed this analysis for the most known semi-Naive Bayes classifiers and decision tree based classifiers. The ensemble of classifiers is discarded because these are based on building multiple models, which causes an excessive overload in the demand for computational resources.

Thus, we divided the classifiers evaluated in this study into two classes. An initial class in which the memory space can be computed a priori, that is to say, when the number of parameters directly depends on the dimensionality of the random variables and/or the size of the dataset. And a second class, in which the number of parameters also depends on the particular joint probability distribution of those variables (i.e. classifiers with a feature selection mechanism or decision trees with a post-pruning process).

Let us introduce some previous notations: the number of classes to be predicted is denoted as $k$; the number of predictive attributes of the data set as $n$; and the mean number of cases of these attributes as $v$.

### 3.3.1 Data independent memory space classifiers

As was stated above, the memory space or memory complexity of these classifiers can be estimated *a priori*, prior to the training stage, having established the parameters $k$, $n$ and $v$. Given the data, this has the advantage of informing about the memory requirements of this classifier.

Not all the classifiers whose memory complexity is independent from the data are included here. We analyzed the best known ones, because of their simplicity or their high performance. Lazy classifiers, such as LBR [196], are discarded because they need to store the complete data set, and therefore they do not perform any data compression at training time.

Although we previously introduced most of these classifiers in Chapter 2, we will enumerate them once again, because in this section we wish to provide an overview that focuses more the memory space analysis of these models.

**Naive Bayes (NB) [54]:** This classifier dramatically reduces its space complexity through the assumption of attribute independence given the class. NB needs a one-dimensional array of class probability estimates and a two-dimensional array of the conditional probabilities of each attribute given each class.

- The classification space complexity is $O(knv)$
- The classification time complexity is $O(kn)$.
- The exact number of parameters is $knv + k$.

**TAN [67]:** This is in the family of one-dependence classifiers: each attribute depends on the class and on another attribute. Once the classifier has been learnt by the maximum spanning tree approach proposed by [67], TAN needs to store the probability estimates of each attribute conditioned to its attribute parent and the class, except for the root attribute of the tree, which is only conditioned to the class.

- The classification space complexity is $O(knv^2)$.
- The classification time complexity is $O(kn)$.
- The exact number of parameters is $knv^2 + kv + k$.

**AODE [176]:** This classifier is based on averaging one-dependence models and has received much attention from the machine learning community. It builds $n$ different one-dependence models, where one attribute is the parent of all the rest, each time placing a different parent attribute. Each model consumes $O(knv^2)$.

- The classification space complexity is $O(kn^2v^2)$.
- The classification time complexity is $O(kn^2)$.

- The exact number of parameters is $kn^2v^2 + knv + k$.

**WAODE [97]:** An extension of AODE which makes a weighted average of the $n$ one-dependence models instead of the simple average of its predecessor. A new weight must therefore be stored, which gives rise to:

- The classification space complexity is $O(kn^2v^2)$.
- The classification time complexity is $O(kn^2)$.
- The exact number of parameters is $kn^2v^2 + knv + n + k$.

**HNB [193]:** This classifier creates a hidden parent for each attribute, which combines the influences from all other attributes. This is translated in terms of space complexity to an estimation of the conditioned probability of an attribute, given another attribute and the class variable for each attribute pair, which involves $kn^2v^2$ parameters, and a weight that measures the normalized mutual information among both attributes, and which involves other $n^2$ parameters.

- The classification space complexity is $O(kn^2v^2)$.
- The classification time complexity is $O(kn^2)$.
- The exact number of parameters is $kn^2v^2 + n^2 + k$.

### 3.3.2 Data dependent memory space classifiers

In the previous class, the big O notation provides direct information on the memory space required by the classifiers. The worst case was close to the mean. But in this class, this agreement does not occur. The worst case or the derived approximation of the big O notation is usually quite distant from the final memory space required by the classifier. A clear example, as we pointed out before, is the case of decision trees.

In this section, we disclose for each one of the examined classifiers its worst and its best possible memory space requirements. The aim of this description it to highlight the wide memory space range to which most of these classifiers are fitted.

**Selective Naive Bayes Classifiers:** Inside this family, different approaches can be found (Section 2.2.2). These are based on a score+search approach which aims to reduce the feature space, removing irrelevant and redundant variables that deteriorate the performance of the Naive Bayes classifier. Their space complexity is the same as NB, $O(knv)$, although the final number of parameters clearly depends on the number of relevant attributes in the database.

- Space Complexity: $O(knv)$
- Minimum Bound: $k$, if no attribute is relevant for class prediction.
- The classification time complexity will depend on the final number of selected variables and will be linear in relation to this number.

For this category the wrapper approach by Kohavi and John [109]is selected (for details, see Section 2.2.2).

**Selective and Joining Naive Bayes Classifiers:** This family was introduced by Kononenko [112] with little success, but later improved upon by Pazzani [137] (see Section 2.2.3 for details). They enable Cartesian products of interdependent Attributes to be created. Its space complexity is conditioned by the size of the Cartesian products created in the learning process. In the worst case all variables can be joined in only one group:

- Space Complexity: $O(kv^n)$
- Minimum Bound: $k$, if no attribute is relevant for class prediction.
- The classification time complexity will be linear in the final number of formed groups.

The *forward sequential selection and joining* version of Pazzani's approach [137] was selected for the evaluation (for details, see Section 2.2.3).

**Decision Tree based Classifiers:** Decision tree based classifiers (Section 2.3.1) are one of the best known and most exploited classification models. Roughly speaking, they are based on a recursive partition of the data space, and use

a tree structure with attributes in its nodes. At each partition, a probability distribution for the class is determined. Thus, the number of possible leaves is $O(t)$ (one leaf per sample) and, in each leaf, a conditional probability for the class is computed. The application of post-pruning techniques can strongly reduce the number of leaves, and therefore the final number of parameters also greatly depends on the data:

- Space Complexity: $O(kt)$
- Minimum Bound: $k$, if no attribute is relevant for class prediction.
- The classification time will be linear in the depth of the final tree after the post-pruning process.

To evaluate this category, we will use the J48 approach, an advanced version of the C4.5 approach by Quinlan [149].

## 3.4 Algorithm Comparisons

In the previous Section 3.2, we introduced a new semi-Naive Bayes proposal, and a space complexity analysis of a wide range of classifiers was detailed in Section 3.3. It can be seen that the group of *data dependent memory space classifiers* presents a wide range to which their memory requirements are fitted. Thus, this experimental section attempts to provide deeper insights about the exact demanded number of parameters of these classifiers and, if where this is not possible, establish a ranking with the most memory-expensive and the least memory-expensive predictors.

Firstly, we establish the settings of the experimental configuration. Subsequently, we present the results of the memory space comparisons. We also include another subsection to highlight the fact that a strong memory space reduction of **SNB-G** does not presuppose significant losses in the classification performance.

### 3.4.1 Experimental setup

For these experiments we selected a wide range of the 33 different datasets taken from the UCI repository. We believe they define a good set of benchmarks for the

Table 3.8: Data Bases Description

| Name | $t$ | $n$ | $k$ | $v$ | Name | $t$ | $n$ | $k$ | $v$ |
|---|---|---|---|---|---|---|---|---|---|
| anneal | 898 | 39 | 6 | 3.4 | kr-vs-kp | 3196 | 37 | 2 | 2.1 |
| audiology | 226 | 70 | 24 | 2.5 | labor | 57 | 17 | 2 | 2.1 |
| autos | 205 | 26 | 7 | 4.3 | lymphography | 148 | 19 | 4 | 2.8 |
| balance-scale | 625 | 5 | 3 | 2.2 | mushroom | 8124 | 23 | 2 | 5.5 |
| wisconsin-cancer | 699 | 10 | 2 | 3.0 | primary-tumor | 339 | 18 | 22 | 3.3 |
| horse-colic | 368 | 23 | 2 | 2.9 | segment | 2310 | 20 | 7 | 8.8 |
| credit-rating | 690 | 16 | 2 | 3.4 | sick | 3772 | 30 | 2 | 2.1 |
| german-credit | 1000 | 21 | 2 | 3.2 | sonar | 208 | 61 | 2 | 1.4 |
| pima-diabetes | 768 | 9 | 2 | 2.1 | soybean | 683 | 36 | 19 | 3.3 |
| Glass | 214 | 10 | 7 | 2.9 | splice | 3190 | 61 | 3 | 4.8 |
| cleveland-diseas | 303 | 14 | 2 | 2.2 | car | 1728 | 7 | 4 | 3.6 |
| hungarian-diseas | 294 | 14 | 5 | 2.3 | vehicle | 846 | 19 | 4 | 3.9 |
| heart-statlog | 270 | 14 | 2 | 1.7 | vote | 435 | 17 | 2 | 2.0 |
| hepatitis | 155 | 20 | 2 | 1.9 | vowel | 990 | 12 | 11 | 4.3 |
| hypothyroid | 3772 | 30 | 4 | 2.4 | waveform | 5000 | 41 | 3 | 3.1 |
| ionosphere | 351 | 35 | 2 | 3.9 | zoo | 101 | 18 | 7 | 7.8 |
| iris | 150 | 5 | 3 | 2.8 | Range | 57-5k | 5-70 | 2-24 | 1.4-8.8 |

objective of this study, as most were extracted from practical applications in very different fields: credit rating, industrial applications, medical applications, etc. Therefore, they will probably be similar to potential applications of supervised classification systems for embedding in memory-limited devices.

In Table 3.8, the datasets with their basic features are listed. In the final row, we display the range of each feature of the datasets with the aim of showing the heterogeneity of this benchmark.

We implemented this approach, the semi-NB with grouping of cases and the "forward sequential selection and joining" version of the semi-Naive Bayes approach by Pazzani [137] in Elvira environment [39]. The experiments, along with the remaining evaluated classifiers, were performed in Weka platform [185].

The implementation of the classifiers Naive Bayes, AODE, TAN, HNB and WAODE were the Weka ones with the default settings. For the decision-tree-based classifiers, we used the implementation of Weka with default settings for J48, an advanced version of Quinlan's C4.5 [149]. For the wrapper selective naive Bayes, we employed the implementation of Weka with a "Best First Search" and default settings.

The data were preprocessed made with the Weka filters themselves, depending on the requirements of each classifier. Basically, J48 did not require any preprocessing and, for the remaining ones, the missing values were replaced (with the

mean value for continuous attributes and with the mode for the discrete ones) and discretized with the Fayyad and Irani method [58].

We used two performance measures to evaluate the performance of a classifier: the classical prediction accuracy (noted as %); and the logarithm of the likelihood of the true class: *Log-Likelihood*= $\ln(P(\hat{c}_i|\mathbf{x}))$, noted as $LL$. This is a more general evaluation of the performance of a classifier, as any loss function will be directly or indirectly based on the precision of these estimates.

The evaluation scheme of the classifiers was performed using a 10-fold-cross validation repeated 10 times for each database. Thus, 100 train and test evaluations were extracted. For each one of these evaluations, we computed the accuracy and the Log-likelihood measures, as well as the number of parameters used by each classifier. In the tables we reported the mean value of these 100 evaluations.

Comparison of these performance measures was made following the methodology proposed by Demsar [47] for comparison of several classifiers over several datasets. With this methodology, the non-parametric Friedman test was used to evaluate the rejection of the hypothesis that all the classifiers perform equally well for a given significance $\alpha$ level (5% in this case). When the Friedman test detects significant differences, a post-hoc test is also used to assess particular differences among these classifiers. The Bonferroni-Dum test [47] was employed with a 5% of significance level fixing *SNB-G* as a reference model. This test assesses significant improvements or degradations, measuring the differences between the average rankings of the classifiers. A threshold is previously set and depends on the number of classifiers being compared; if these differences are higher than this threshold, there is a statistical improvement or degradation, depending on the sign of the difference.

### 3.4.2 Memory space comparison

As in the present study we aimed to evaluate which classifier family is the most suitable for integration into memory-limited devices, we first give a detailed view of the memory requirements of the nine classifiers analyzed.

Table 3.9: Number of Kilobytes of memory needed to define the classification models

| | NB | SNB-G | WNB | J48 | WSNB | TAN | AODE | HNB | WAODE |
|---|---|---|---|---|---|---|---|---|---|
| Average | 3.9 | 2.5 | 1.8 | 1.9 | 26.1 | 15.2 | 518.9 | 525.4 | 519.1 |
| Desv | 6.4 | 4.8 | 2.7 | 2.6 | 80.0 | 23.3 | 1108.3 | 1114.7 | 1108.4 |
| Minimum | 0.3 | 0.2 | 0.1 | 0.1 | 0.2 | 0.6 | 2.9 | 3.1 | 2.9 |
| Maximum | 33.4 | 21.8 | 10.8 | 9.1 | 463.8 | 84.7 | 5927.4 | 5965.7 | 5928.0 |

(The full expanded table can be found in Appendix: Table 1)

We employed the following procedure to compute the memory space of each predictor: for the data independent group, we computed the memory space using the derived formulas of Section 3.3; for the data dependent group we estimated this space as the mean of the values computed in each training and test evaluation. Assuming that for each parameter we need a double precision number, i.e., 8 bytes, Table 3.9 shows the statistics of the Kilobytes consumed by each classification model. Mean number and deviation are provided, along with the minimum and maximum number of Kilobytes required for some of the 33 datasets evaluated.

Observing the different rows of this Table 3.9, we can extract several conclusions. Excluding WSNB, data-dependent classifiers need less memory in mean (a small number of Kilobytes) than data-independent ones (hundreds or tens of Kilobytes). Furthermore, the deviation, the minimum and the maximum of the ratios are strictly lower for data-independent classifiers. Special attention must be paid to the fact that a simple classifier such as the Naive Bayes consumes more memory resources than WNB, J48 and SNB-G, which attempt to model dependencies among attributes.

Only TAN predictor presents a memory demand similar in scale to data-dependent classifiers and Naive Bayes. The memory requirements of AODE, HNB and WAODE stand out clearly, in some cases overcoming the Megabyte threshold (for details see Table 1 in the Appendix).

The data in Table 3.10 were obtained by setting the SNB-G approach as the reference predictor and measuring the memory complexity of the remaining classifiers as their ratio between their respective memory loads in each dataset. Readers should be aware that the mean of theses ratios is not the ratio of the

Table 3.10: Memory space ratio respect to SNB-G
(The full expanded table can be found in Appendix: Table 2)

| | NB | WNB | J48 | WSNB | TAN | AODE | HNB | WAODE |
|---|---|---|---|---|---|---|---|---|
| Mean | 2.4 | 1.0 | 1.7 | 21.5 | 8.5 | 217.0 | 226.3 | 217.3 |
| Deviation | 1.8 | 0.6 | 2.9 | 69.1 | 9.6 | 265.2 | 271.5 | 265.4 |
| Minimum | 0.8 | 0.4 | 0.2 | 0.4 | 2.7 | 14.2 | 14.8 | 14.4 |
| Maximum | 9.5 | 3.4 | 14.7 | 399.1 | 52.4 | 1205.6 | 1225.4 | 1206.5 |

means of Table 3.9. Here, in Table 3.10, the average value across the 33 datasets is given along with the deviation, the minimum and the maximum of the ratios in any of these databases (the fullly expanded table can be found in the Appendix, Table 2).

In Table 3.10, the "Average" row shows that the SNB-G is on average less memory-demanding than the remaining approaches, due to the fact that their averages are greater than the unit. Only Wrapper Naive Bayes exhibits a similar value. Observing the 'Minimum" and "Maximum" rows, we can see than the range of the ratios is much more balanced in favour of the SNB-G approach, as the margins are higher when this predictor wins than when it loses (for a close review look Table 2 in the Appendix).

In short, data-dependent classifiers are most effective with regard to compressing the data distribution. Thus, they are more suitable for embedding in memory-limited devices, although the NB and TAN approaches also appear to be suitable for this purpose, because the range of their memory requirements is also low.

### 3.4.3 Classification performance comparison

Once we have established the most effective classifiers under a memory load point of view, in this subsection we will see whether this memory efficiency is associated with a substantial deterioration of the classifier performance.

Table 3.11 summarizes the average performance of the nine classifiers. The "B/L" rows provides the number of datasets in which each classifier presents a higher (H) or lower (L) degree of accuracy or log-likelihood mean value in relation to SNB-G. In general terms, there are no big differences in terms of accuracy. In terms of log-likelihood of quality of estimates of class probability, there are higher

differences in favour of SNB-G versus NB, WNB, J48 and WSNB, whereas AODE, HNB and WAODE behave better.

Table 3.11: Accuracy and log-likelihood performance

| Dataset | SNB-G | NB | WNB | J48 | WSNB | TAN | AODE | HNB | WAODE |
|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 83.89 | 82.67 | 83.3 | 83.93 | 79.6 | 84.54 | 84.72 | 84.96 | 85.47 |
| H/L | | 15/18 | 13/20 | 18/15 | 9/17 | 23/10 | 20/13 | 24/9 | 27/6 |
| Log-like. | -0.69 | -0.99 | -0.71 | -0.79 | -0.85 | -0.74 | -0.70 | -0.69 | -0.67 |
| H/L | | 5/28 | 9/24 | 11/22 | 8/18 | 16/17 | 19/14 | 22/11 | 21/12 |

(The full expanded tables can be found in Appendix: Table 3 and Table 4)

Following the methodology described at the end of subsection 3.4.1, SNB-G is compared with those semi-NB classifiers based on the mixture of models and with a low memory efficiency: AODE, WAODE and HNB. The results of this test are shown in Table 3.12 (the lower the ranking, the better the classifier performs).

The first conclusion from this table is that the SNB-G approach is just under the performance of this advanced classifier in terms of accuracy. Only WAODE is statistically better, the remaining ones being just over SNB-G, but not showing a significant level. When the quality of the probability estimates, $LL$, is compared, SNB-G is found to be more competitive and the Friedman Test shows no differences among the performance of these classifiers.

Table 3.12: Performance comparison with low memory efficient classifiers.

| Ranking | SNB-G | AODE | HNB | WAODE |
|---|---|---|---|---|
| Accuracy | 3.1 | 2.6 | 2.6 | $1.7^{\top}$ |
| Log-like. | 2.9 | 2.7 | 2.2 | 2.2 |

$\top$ statistically significant improvement respect to SNG-G.

In this second round, the SNB-G predictor was compared with classifiers presenting a similar memory efficiency; NB, TAN, J48 and WNB (results in Table 3.13). Once again, the conclusions can be easily deduced: they all perform similarly in terms of accuracy (only the TAN approach stands out a little, but not significantly) and SNB-G has much better probability class estimates with sta-

tistically significant differences with NB, WNB and J48, whereas no difference is found with respect to the TAN predictor.

Table 3.13: Performance comparison with high memory efficient classifiers.

| Ranking | SNB-G | NB | WNB | J48 | TAN |
|---|---|---|---|---|---|
| Accuracy | 3.1 | 3.1 | 3.5 | 3.0 | 2.3 |
| Log-like. | 2.3 | $3.9^{\perp}$ | $3.4^{\perp}$ | $3.4^{\perp}$ | 2.2 |

$\perp$ statistically significant degradation respect to SNG-G.

## 3.5 Conclusions and Future Work

In this chapter, we present a new semi-Naive Bayes classifier with grouping of cases which achieves a competitive level of performance, particularly in terms of quality of class probability estimates, and with a high memory efficiency. The memory space complexity of some of the best known state-of-the-art probabilistic classifiers was also studied.

The focus of this analysis was two-fold:

- Firstly, to propose a new semi-Naive Bayes approach that exploits the grouping of cases of new compound variables intended to boost the performance classification and to reduce the complexity of the model to make it more efficient with respect to time and memory requirements. To this end, we proposed and evaluated three different joining and grouping criteria (Section 3.2).

- Secondly, the another objective of this chapter was to determine which classifiers exhibit the best trade-off in terms of performance and memory efficiency. it can be seen that J48, TAN and SNB-G present an excellent compromise between accuracy, log-likelihood and memory demand. Concretely, SNB-G presents the best trade-off between memory efficiency and quality of class probability estimates, together with a competitive level of accuracy.

Mainly, we have shown that use of a *grouping of cases* approach can maintain the performance of a classifier while reducing the number of required parameters. This technique could be further applied to some of the classifiers studied in this dissertation, particularly those presenting very good performance, but with high memory requirements, for example the AODE [176] classifier, which could lead to a much more memory-efficient model with a highly competitive level of performance.

# Chapter 4

# A Bayesian account of classification trees

In this chapter, we present a Bayesian account of the problem of inferring classification trees. Concretely, in Section 4.3, we address the problem of estimating class probabilities using a smoothing approach that attempts to simulate a post-pruning process. In Section 4.4.3, we deal with the problem of constructing several classification tree models, building an ensemble of trees. In both cases, Bayesian inspired approaches are applied.

## 4.1 Motivation

Decision trees or classification trees (decision trees in which a probability of class membership, rather than simply the class label, is predicted) are the predictive models most commonly employed and studied (for details see Section 2.3).

Furthermore, the ensemble of decision trees (for details, see Section 2.3.2) arose some years later, as an extension of decision tree models, in an attempt to boost the performance of single classification trees, while maintaining some of their main properties. The basic idea consists of generating a set of different decision trees, combining them with a majority vote criteria. That is to say, when an unlabeled unclassified instance arises, each single decision tree makes a prediction and the instance is assigned to the class value with the highest number of votes. Shapire's Adaboost [64] and Breiman's Random Forests [26] stand out

as state-of-the-art classification models and both are based on the idea of decision tree ensembles.

This chapter aims to tackle the problem of inferring single classification trees and ensemble of classification trees using Bayesian inspired approaches but without following a strictly full Bayesian approach. *Bayesian model selection* [174] and *Bayesian model averaging* [83] are the basis of our approach.

## 4.2 Bayesian Inference of Classification Trees

### 4.2.1 Basic Framework

In order to introduce the basic framework for applying the Bayesian approach for inferring classification trees, the notation used by Buntine [30] is followed. Buntine was the first author to apply Bayesian techniques to this specific problem.

Classification trees partition the space of examples into disjoint subsets, each one represented by a leaf in the tree, and associates a conditional probability distribution for the class variable in relation to the configuration that defines the partition assigned to that leaf.

It is assumed that there are $K$ mutually exclusive and exhaustive classes, $c_1, ..., c_K$. Assuming that example $x$ falls to leaf $l$ in the tree structure $T$, then the tree gives a vector of class probabilities $\phi_{k,l}$ for $k = 1, ..., K$, which are the probability of class $c_k$ at leaf $l$. Thus, a classification tree has a discrete component determined by the structure of tree $T$ and a continuous component that is given by the class probabilities of all the leaves of the tree $\Phi_T = \{\phi_{k,l} : k = 1, ..., K; l \in leaves(T)\}$. No more parameters are needed, as it is assumed that all variables are multinomial, although continuous variables could be also managed, including the cut-points in the branching nodes.

Thus, for the above mentioned example $x$ falling into leaf $l$, its predicted probability class value is described as $P(C = c_k|x, T, \Phi_T) = \phi_{k,l}$. If a concrete class had to be predicted, it would be the one with the highest probability.

Under the Bayesian approach, the quality of the models is evaluated as their posterior probability, given the learning data. This learning data comprises a set

of $N$ i.i.d. samples, $(\bar{c}, \bar{x}) = \{(c_1, x_1), ..., (c_N, x_N)\}$. The probability of the model can be computed using the Bayes's theorem as:

$$P(T, \Phi_T|\bar{c}, \bar{x}) \propto P(T, \Phi_T|x) \prod_{i=1}^{N} P(c_i|x_i, T, \Phi_T) = P(T, \Phi_T|x) \prod_{l \in leaves(T)} \prod_{k=1}^{K} \phi_{k,l}^{n_{k,l}} \quad (4.1)$$

where $n_{k,l}$ is the number of samples of class $c_k$ falling into leaf $l$. $P(T, \Phi_T|x)$ can be considered equal to $P(T, \Phi_T)$ as the prior over the models, as $T$ and $\Phi_T$ are conditioned to unclassified samples.

The factor $\Phi_T$ can be removed from Equation 4.1 if a prior over the set of parameters is defined and integrated into them. This can be easily achieved if the conjugate of this prior has the same functional form, as it is the case of Dirichlet distributions.

**Parameters Priors:** It is assumed that the prior beliefs over parameter values are given by a Dirichlet distribution. It is also assumed that these distributions are independent from the parameters of the different leaves of the tree. That can be formulated as follows:

$$P(\Phi_T|T) = \prod_{l \in leaves(T)} \frac{1}{B_K(\alpha_1, ..., \alpha_K)} \prod_{k=1}^{K} \phi_{k,l}^{\alpha_{k,l}-1}$$

where $B_K$ is the K-dimensional beta function and $\alpha_i$ are the parameters of the Dirichlet. $B_K$ is computed in terms of product of gamma functions $\Gamma(x)$ ($\Gamma(x+1) = x\Gamma(x)$) as follows:

$$B_K(x_1, ..., x_K) = \frac{\prod_{i=1}^{K} \Gamma(x_i)}{\Gamma(\sum_{i=1}^{K} x_i)}$$

**Posterior Tree Probability:** Therefore, using these priors, the posterior probability of a tree, $T$, can be computed as follows:

$$P(T|\bar{x}, \bar{c}) \propto P(\bar{c}|\bar{x}, T)P(\bar{x}|T)P(T) = P(\bar{x}|T)P(T) \int_{\Phi_T} P(\bar{c}|\bar{x}, T, \Phi_T)P(\Phi_T|T)d\Phi_T$$

This integral can be computed using the above formulation. At the same time, $P(x|T)$ is included in the proportional constant, as it is assumed to be the same for all the tree structures.

$$P(T|\bar{x},\bar{c}) \propto P(T) \prod_{l \in leaves(T)} \frac{B_C(n_{1,l}+\alpha_1,...,n_{K,l}+\alpha_K)}{B_C(\alpha_1,...,\alpha_K)} \tag{4.2}$$

Although Buntine tested several priors over the possible tree structures, $P(T)$, in an attempt to favour simpler trees, there was no definitive recommendation [30]. A uninform prior over the possible tree structures will therefore be assumed.

**Posterior Class Probability Estimates:** Finally, the estimations of the probabilities of the leaves of the tree $T$ are also computed by averaging all possible parameter configurations, by means of expectation:

$$P(C=c_k|x,T,\bar{c},\bar{x}) = \int_{\Phi_T} \Phi_{k,l} P(\Phi_T|T,\bar{c},\bar{x}) d\Phi_T = \frac{n_{k,l}+\alpha_j}{n_k+\alpha_0} \tag{4.3}$$

where $l$ is the leaf in which $x$ falls and $n_l = \sum_{k=1}^{K} n_{k,l}$ and $\alpha_0 = \sum_{k=1}^{K} \alpha_k$.

### 4.2.2 Single Classification Trees

Single trees are built using the same recursive approach from the root node to the leaves (for details, see Section 2.3.1). Thus, at any node of the tree, $S_0$, there is a vector of counts of the samples across the different classes $V_{S_0} = (n_1,...,n_K)$ and a set of possible split attributes $(S_1,...,S_p)$. It remains to be decided whether vector $V_{S_0}$ is split with some candidate attribute $S_i$ in order to make a better estimate of the conditioned class probabilities. Thus, vector $V_{S_0}$ would be replaced by a set of $|S_i|$ (cardinality of $S_i$ attribute) vectors generated by the new partitions induced by the inclusion of $S_i$. Let us denote as $T_{S_0}$ the former tree and $T_{S_i}$ the new tree with the inclusion of the split attribute $S_i$.

For each possible tree $\{T_{S_1},...,T_{S_p}\}$ a score is computed, $Score(T_{S_i}) = logP(T_{S_i}|\vec{c},\vec{x})$, using Equation (4.2). If $Score(T_{S_i}) - Score(T_{S_0}) > 0$ then $S_i$ must be included in the tree. This difference can easily be computed because the

probability of the tree is multiplicative across the leaves, and the local difference of the scores of the leaves is therefore sufficient to evaluate the condition [30].

If none of the candidate split attributes shows a better score, the tree growing must stop at $S_0$, but if there are several $S_i$ with higher scores, which one should be used to split?. When only one tree is sought, as in this case, a greedy approach is usually employed [27; 30; 150], selecting at each step the split that produces the highest score. That is to say, the split attribute $S^*$ is defined as follows:

$$S^* = argmax_{S_i}\{Score(S_i) - Score(S_0) \geq 0\}$$

When information-based scores are used to grow classification trees such as Information Gain [150] or Gini Index [27], they always predict better partitions whenever a new split node is added (differences are always greater or equal than zero). Thus, stop criteria usually include conditions such as a minimum threshold for the number of samples, or a pure partition of the data to prevent against an excessive branching. However, this kind of corrections were insufficient, and post-pruning techniques were therefore applied for further simplification of the inferred tree structure [148]. With Bayesian metrics this problem is partially solved because of the inherent penalty they provide to more complex models (i.e. excessive branching): it is possible to have $Score(S_i) - Score(S_0) < 0$.

### 4.2.3 Multiple Classification Trees

In the full Bayesian approach, inference considers all possible models with the corresponding posterior probability and not just the most probable one. In order to handle several models, the final prediction is performed by adding each particular prediction of each model weighted by its posterior probability:

$$P(C = d_j|x, \bar{c}, \bar{x}) = \sum_T \int_{\Phi_T} P(C = d_j|x, T, \Phi_T) P(T, \Phi_T|\bar{c}, \bar{x}) d\Phi_T \tag{4.4}$$

where the summation covers all possible tree structures.

In Bayesian Model Averaging, Equation (4.4) is approximated by using importance sampling and Monte-Carlo estimation. Thus, tree structures will be

generated in an approximate proportion to their posterior probabilities. But applying Monte-Carlo methods in this huge model space would lead to a very computationally expensive approach.

Buntine computed two approximations to this sum by reducing the set of tree structures [30]. One approximation, known as **Smoothing**, restricted the structures to the ones obtained by pruning a complete tree. It is a smoothing because probabilities at final leaves are computed by averaging them with some of the class probabilities from the interior nodes of the tree. The other approximation used by Buntine was called **Option Trees** [30]. This approximation was based on searching and storing many dominant terms of the sum, i.e., trees with high posterior probabilities. The multiple tree structures were compactly represented using AND-OR nodes. The final predictions were made by averaging the predictions of the different models encoded in these option trees.

Other studies [49] have attempted to apply a Bayesian model averaging approach for weighting each single tree by approximation of their posterior probabilities (they followed the scheme of Equation (4.4)). Therefore, rather than a simple majority vote as the previous ensembles of trees approaches, they returned a weighted averaged of the single predictions. But this approach did not provide good results.

## 4.3 A Bayesian approach to estimate probabilities in classification trees

In this section, we present a new Bayesian method for estimating the probability of class membership. The procedure is based on a Bayesian approach that weights different rules of the induced tree in an attempt to simulate a post-pruning process. In an experimental evaluation, we demonstrate that this approach reaches the performance of J48 (an advance version of Quinlan's C4.5), one of the best known decision tree inducers, in terms of predictive accuracy, and outperforms it in terms of better probability class estimates.

### 4.3.1 Introduction

One of the main problems of classification trees is the poor estimates of class probabilities they usually provide [24; 138]. In many situations, a good estimate of the probability class is an obvious requirement, particularly when we need a ranking of the samples according to the class they belong to. (i.e., most of the web search engines rank the web pages based on their probability of being relevant for a given query following the *Probability Ranking Principle* [151]).

In [146] a survey study of different methods for better probability of class estimates was performed and was based on C4.5. They compare three different methods: Laplace estimate, C4.5 pruning [149] and, particularly, bagging [23]. They conclude with positive evidence in favour of Laplace and Bagging, but do not reach a definitive conclusion with regard to pruning.

In this section, we present a Bayesian approach to inducing classification trees. The aim is to maintain the predictive accuracy of one of the state-of-the-art classification tree inducers $J48$ (an advanced version of Quinlan's C4.5) and to make significant improvements in the estimates of probability class beyond the use of Laplace correction or a post-pruning process. In order to demonstrate this, an experimental study with 27 UCI databases to evaluate this approach was conducted.

The rest of the section is organized as follows. In Section 4.3.2 the proposed smoothing method for the class probabilities of the tree leaves is introduced. Subsequently, this smoothing is improved with the definition of non-uniform priors over the parameters of the tree, Section 4.3.3. Finally, the experimental evaluation and the results are shown in Section 4.3.4.

### 4.3.2 Bayesian Smoothing approach to estimate class probabilities

This class probability smoothing approach can be seen as an intermediate approach between a single classification tree and a multiple classification tree model. The smoothing approach presented in this section attempts to change the estimation derived from Equation 4.3 in Section 4.2.1 with a weighted average that

considers the partial estimations made by the internal nodes from the smoothed leaf to the root node.

The criteria used to grow the classification tree (Section 4.2.2) attempt to select the most probable model (under the considered assumptions) branching when there is a positive difference between the scores of the models. Sometimes the differences between these scores may be small (mainly when the branching is in the final steps) and the tree resulting from the ramification can be as plausible as the smaller tree without further ramifications. This leads us to a multiple model problem as was described in Section 4.2.3. The post-pruning problem in decision trees is a very similar issue, and attempts to simplify the tree without further deterioration in its approximation capacity.

The approach described herein attempts to tackle this issue by averaging the predictions of the internal nodes of the tree in the path from the root node to the leaf that is being smoothed. This average is weighted. Let us denote as $S_0$ the node that is evaluated, either as a leaf node or being further branched by introducing a new split node $S_i$. The difference between the scores of both nodes (for details, see Section 4.2.1) will be used to weight the strength that is to be associated to the partial prediction made by $S_0$. If this difference is very high in favour of $S_i$, the weight associated with the prediction of $S_0$ will be negligible. To the contrary, if this difference is minute, this weight will be much higher.

Let us formally define how the probabilities at the leaf nodes are estimated with this Bayesian smoothing approach. As is shown in Equation 4.3, the posterior probability of class $C$, given an instance, $x$, a tree model $T$ and learning data $(\vec{c}, \vec{x})$ is computed as follows:

$$P(C = c_k|x, T, \bar{c}, \bar{x}) = \frac{n_{k,l} + \alpha_{l_i,k}}{n_l + \alpha_{l_i}}$$

where $l$ is the leaf in which $x$ falls and $n_l = \sum_{k=1}^{K} n_{k,l}$ and $\alpha_0 = \sum_{k=1}^{K} \alpha_k$.

The idea of considering several tree models, as depicted in Section 4.2.3, is employed in this case. This class probability is therefore estimated as follows:

$$P(C = c_k|x, T, \bar{c}, \bar{x}) \propto \sum_{l_i \in Path(x,T)} \frac{n_{k,l_i} + \alpha_k}{n_{l_i} + \alpha_0} \mathcal{W}(l_i, T)$$

where $Path(x, T)$ is the set of internal nodes (but which are supposed to act as leaves) in the path of $T$ from the leaf where $x$ falls in $T$ up to the root node. $\mathcal{W}(l_i, T)$ is the weight associated with the prediction in node $l_i$ and is defined as follows:

$$W(l_i, T) = \prod_{l_j \in Path(l_i,T)} \frac{B_C(\alpha_{1,l_{j-1}}, ..., \alpha_{K,l_{j-1}})}{B_C(n_{1,l_{j-1}} + \alpha_{1,l_{j-1}}, ..., n_{K,l_{j-1}} + \alpha_{K,l_{j-1}})} \cdot \tag{4.5}$$

$$\cdot \prod_{h_i \in Children(l_{j-1})} \frac{B_C(n_{1,h_i} + \alpha_{1,h_i}, ..., n_{K,h_i} + \alpha_{K,h_i})}{B_C(\alpha_{1,h_i}, ..., \alpha_{K,h_i})} \tag{4.6}$$

$$\tag{4.7}$$

where $Path(l_i, T)$ is the set of nodes in the path of $T$ from the node $l_i$ to the root node; $l_{j-i}$ is considered the ancestor node of $l_j$ in the previous path; and $Children(l_{j-1})$ is the set of children of this ancestor $l_{j-1}$. Finally, a normalization is required.

As can be seen, this weight is computed as an accumulated product of the product of the exponential differences between the scores of a node in relation to its ancestor. The bigger the difference, the greater the weight and the greater the impact of the internal estimation on the final class probabilities of the smoothed leaf.

This approach also has the advantage that all these probabilities can be efficiently computed as the tree is built, with only a linear increase in the complexity.

In order to clarify this approach, we show in Algorithm 7 a pseudo-code description of the Bayesian smoothing approach of this section. The part related to the selection of the split node is omitted in order to not introduce complexity in this description. However, as was previously mentioned, this smoothing can be performed as the tree is generated.

**Algorithm 7** *Bayesian Smoothing Algorithm*

***BayesianSmoothing(*** $T$**,** $l_j$**,** $C = \{c_1, ..., c_k\}$**,** $W$***)***

$\{n_{1,l_j}, ..., n_{k,l_j}\}$=*Samples(*$T$*,* $l_j$*);*

$c_k = c_k + \frac{n_{1,l_j} + \alpha_{1,l_j}}{n_{l_j} + \alpha_{l_j}} \cdot W$ *for* $i = 1, \ldots, K$.

*if* $l_j$ *is a leaf*

*normalize(C);*

*setSmoothedClassProbabilities(*$l_j$*,C);*

*else*

*total=0;*

*for each child* $h_i$ *of* $l_j$ *in T;*

$\{n_{1,h_i}, ..., n_{k,h_i}\}$=*Samples(T,* $h_i$*);*

$total = total + log(\frac{B_C(n_{1,h_i}+\alpha_{1,h_i},...,n_{K,h_i}+\alpha_{K,h_i})}{B_C(\alpha_{1,h_i},...,\alpha_{K,h_i})})$;

*end-for*

$total = total - log(\frac{B_C(n_{1,l_j}+\alpha_{1,l_j},...,n_{K,l_j}+\alpha_{K,l_j})}{B_C(\alpha_{1,l_j},...,\alpha_{K,l_j})})$;

$W = W \cdot e^{total}$;

*for each child* $h_i$ *of* $l_j$ *in T;*

*BayesianSmoothing(* $T$, $h_i$, $C = \{c_1, ..., c_k\}$, $W$*);*

*end-for*

*end-if*

*return;*

### 4.3.3 A Heuristic to define non-Uniform Dirichlet Priors

One of the key assumptions in the previous approach is that the prior distribution of the parameters of the tree model $T$ follows a Dirichlet distribution with parameters $\alpha_k$ $(k = 1, ..., K)$. Common implementations of this approach [80] employ non-informative priors, usually a global sample size, $S$, is assumed and Dirichlet parameters are defined as constants: $\alpha_k = S/K$ $(k = 1, ..., K)$.

This section attempts to present a heuristic to define non-uniform Dirichlet priors which exploits the following concept: if at some node, $S_i$, the frequency $n_k$ is zero, then $\forall j > i$ at $l_j$ descendant nodes, frequency $n_{k,l_j}$ shall also be zero. It therefore makes sense to assume that $n_{k,l_j}$ will probably be zero or close to zero for most future samples $\mathbf{x}$. Thus, reducing the prior probability for $c_k$ at $l_j$

appears to be coherent. Thus, we propose the following heuristic to modify the parameters of the Dirichlet priors distributions:

Let us denote as $\delta_{l_i} = |\{n_{k,l_i} = 0 : k = 1, \ldots, K\}|$ the number of classes with null-frequency in the learning data $D$ restricted to the configuration defined by $l_i$. If $\delta_{l_i} > 1$, the $\alpha_k^{l_{i+1}}$ values are defined as follows:

$$\alpha_{k,l_{i+1}} = \frac{S}{(K - \delta_{l_i} + 1)} \; if \; n_{k,l_i} \neq 0$$

$$\alpha_{k,l_{i+1}} = \frac{S}{(K - \delta_{l_i} + 1)\delta_{l_i}} \; if \; n_{k,l_i} = 0$$

As can be seen, the cases with non-null frequency have the same prior probability, $\frac{1}{(K-\delta_{l_i}+1)}$, while all cases with null-frequency share the same probability mass of one non-null frequency case, i.e., $\frac{1}{(K-\delta_{l_i}+1)\delta_{l_i}}$. It should be pointed out that, with this heuristic, a uniform prior is obtained for a two-class problem. Thus, this heuristic is only effective for multiclass classification problems, because in those cases the null-frequency classes arise much more frequently.

### 4.3.4 Experimental Results

In this section, the experimental evaluation of this approach is presented. Firstly, the evaluation methodology is described and the experimental results of the different approaches are then detailed.

#### Experimental and Evaluation Setup

For these experiments, we selected a set of 27 different datasets taken from the UCI repository. In Table 4.1, the datasets with their basic features are listed. In the last row, we show the range of each feature of the datasets in order to show the heterogeneity of this benchmark.

The classification tree inducers were implemented in Elvira platform [39] and evaluated in Weka [185]. The data were preprocessed and the missing values (with the mode value for nominal attributes and with the mean value for continuous attributes) replaced and discretized with an equal-frequency method with 5 bins using Weka filters themselves.

Table 4.1: Data Bases Description

| Name | $t$ | $n$ | $k$ |
|---|---|---|---|
| anneal | 898 | 39 | 6 |
| audiology | 226 | 70 | 24 |
| autos | 205 | 26 | 7 |
| breast-cancer | 286 | 10 | 2 |
| horse-colic | 368 | 23 | 2 |
| german-credit | 1000 | 21 | 2 |
| pima-diabetes | 768 | 9 | 2 |
| glass2 | 163 | 10 | 2 |
| hepatitis | 155 | 20 | 2 |
| hypothyroid | 3772 | 30 | 4 |
| ionosphere | 351 | 35 | 2 |
| kr-vs-kp | 3196 | 37 | 2 |
| labor | 57 | 17 | 2 |
| letter | 20000 | 16 | 2 |
| | ..... | | |
| lymphography | 148 | 19 | 4 |
| mfeat-pixel | 2000 | 240 | 10 |
| mushrooms | 8123 | 22 | 2 |
| optdigits | 5620 | 64 | 10 |
| segment | 2310 | 20 | 7 |
| sick | 3772 | 30 | 2 |
| solar-flare | 323 | 13 | 2 |
| sonar | 208 | 61 | 2 |
| soybean | 683 | 36 | 19 |
| sponge | 76 | 45 | 3 |
| vote | 435 | 17 | 2 |
| vowel | 990 | 12 | 11 |
| zoo | 101 | 17 | 7 |
| Range | 57-20k | 9-240 | 2-24 |

Two evaluation or performance measures are employed in this experimental evaluation: the classical prediction accuracy (noted as *Accuracy*); and the logarithm of the likelihood of the true class, computed as: *log-likelihood*$= \ln(\hat{P}(c_i|\mathbf{x}))$, where $c_i$ is the true class value of the test example $\mathbf{x}$. The latter score is introduced in order to evaluate the precision of probability class estimates. The usefulness of this score for this task is justified in many ways, as for example in [76; 153].

The evaluation of the classifiers was achieved by a 10-fold-cross validation repeated 10 times for each database. Thus, 100 train and test evaluations are performed.

The comparison of these performance measures was made by following the methodology proposed by Demsar [47] for the comparison of several classifiers over several datasets. With this methodology, the non-parametric Friedman test is used to evaluate the rejection of the hypothesis that all the classifiers perform equally well for a given significance $\alpha$ level (5% in this case). When the Friedman test detects significant differences, a post-hoc test is also used to assess particular differences among these classifiers. The Bonferroni-Dum test [47] was employed with a 5% significance level, setting $J48$ as the reference model. This test assesses significant improvements or degradations, measuring the differences between the average rankings of the classifiers. A threshold is previously set which depends on the number of classifiers being compared; if these differences are higher than

this threshold, a statistical improvement or degradation occurs, depending of the sign the difference.

It is a well-established fact that non-parametric tests impose a stricter condition for rejecting hypotheses. Thus, in this evaluation we display the ranking score that the Friedman test assigns to each classifier (ranking scores close to 1 indicate better a performance for these classifiers) with the idea of detecting some trends, although they may not reach significance levels. These rankings are shown with the label "Ranking".

As previously indicated, the aim of this experimental evaluation is threefold: the tree induction with a Bayesian metric (BM), Section 4.2.2; a Bayesian smoothing approach (BS) for estimating the probabilities class membership, Section 4.3.2; and a non-uniform prior (NUP) definition approach, Section 4.3.3. In all cases, the same prior Dirichlet distribution is used with the same global sample size, $S$. Three different global sample sizes were evaluated: $S = 1$, $S = 2$ and $S = K$.

Let us define the three combinations evaluated: $\beta_S$: classification trees induced with a Bayesian metric (BM); $\widehat{\beta}_S$: BM + BS; $\widehat{\beta}^{\theta}_S$: BM + BS + NUP.

As the aim of this section is to provide a classification tree inducer, based on Bayesian ideas, which reaches the performance of $J48$ (an advance version of Quinlan's C4.5 implemented in Weka platform [185]) in terms of accuracy, but outperforms in terms of better probability class estimates, the three above mentioned approaches are tested by setting $J48$ as a classifier reference.

### Bayesian Metric as splitting critera for inducing CT

We test the use of a Bayesian metric as a splitting criterion for inducing classification trees (CT), previously described in Section 4.2.2.

The results of Table 4.2 show that the use of a Bayesian metric as a splitting criterion is competitive in relation to $J48$. The Friedman test does not reject the hypothesis that all classifiers perform equally well for accuracy and log-likelihood estimates. As can be seen, the mean values are very close to each other.

Therefore, a simple classification tree inducer such as ID3, but with a Bayesian Metric, performs similarly to the one obtained by $J48$, which is much more complicated to implement.

Table 4.2: Bayesian metric as Splitting Criteria
(The full expanded table can be found in Appendix: Table 5 and Table 6)

| Average | $J48$ | $\beta_{S=1}$ | $\beta_{S=2}$ | $\beta_{S=K}$ |
|---|---|---|---|---|
| Accuracy | 85.50 | 85.30 | 85.56 | 84.03 |
| Log-likel. | -0.79 | -0.79 | -0.78 | -0.78 |

| Ranking | $J48$ | $\beta_{S=1}$ | $\beta_{S=2}$ | $\beta_{S=K}$ | Friedman Test |
|---|---|---|---|---|---|
| Accuracy | 2.2 | 2.5 | 2.3 | 3.0 | Accept |
| Log-likel. | 2.3 | 3.0 | 2.3 | 2.4 | Accept |

Table 4.3: Bayesian Smooth Approach
(The full expanded table can be found in Appendix: Table 5 and Table 6)

| Average | $J48$ | $\widehat{\beta}_{S=1}$ | $\widehat{\beta}_{S=2}$ | $\widehat{\beta}_{S=K}$ |
|---|---|---|---|---|
| Accuracy | 85.50 | 85.50 | 85.82 | 83.98 |
| Log-likel. | -0.79 | -0.63 | -0.63 | -0.77 |

| Ranking | $J48$ | $\widehat{\beta}_{S=1}$ | $\widehat{\beta}_{S=2}$ | $\widehat{\beta}_{S=K}$ | Friedman Test |
|---|---|---|---|---|---|
| Accuracy | 2.1 | 2.7 | 2.2 | 2.9 | Accept |
| Log-likel. | 2.8 | 2.5 | 2.0 | 2.7 | Accept |

### Bayesian Smooth Approach

Herein we evaluate the introduction of the Bayesian smoothing approach, $\widehat{\beta}_S$. Results are presented in Table 4.3.

As can be seen, the introduction of the Bayesian smoothing approach involves a slight increment in average accuracy, but an important improvement in terms of quality class probability estimates (log-likelihood). However, the improvement in log-likelihood is not sufficient to cause the rejection of the null hypothesis by the Friedman Test. Nonetheless, the ranking for $\widehat{\beta}_{S=2}$ is now better than the previous model, which excludes the smoothing method.

Table 4.4: Non-Uniform Priors Definition
(The full expanded table can be found in Appendix: Table 5 and Table 6)

| Average | $J48$ | $\widehat{\beta}^{\theta}_{S=1}$ | $\widehat{\beta}^{\theta}_{S=2}$ | $\widehat{\beta}^{\theta}_{S=K}$ |
|---|---|---|---|---|
| Accuracy | 85.50 | 85.85 | 86.04 | 85.37 |
| Log-likel. | -0.79 | -0.61 | -0.60 | -0.69 |

| Ranking | $J48$ | $\widehat{\beta}^{\theta}_{S=1}$ | $\widehat{\beta}^{\theta}_{S=2}$ | $\widehat{\beta}^{\theta}_{S=K}$ | Friedman Test |
|---|---|---|---|---|---|
| Accuracy | 2.4 | 2.6 | 2.3 | 2.7 | Accept |
| Log-likel. | 3.0 | 2.6 | $2.0^{\top}$ | 2.4 | Reject |

$^{\top}$ indicates this classifier is statistically better than $J48$.

### Non-Uniform Dirichlet Priors Definition

Finally, we test the introduction of non-uniform priors in the Bayesian metrics. Results are shown in Table 4.4.

As can be seen, the introduction of non-uniform priors, as detailed in Section 4.3.3, improves both accuracy and log-likelihood. In terms of accuracy, the average is further improved, but it is in the evaluation of the quality of class probability estimates where the best improvements are achieved. Now, the Friedman test rejects the idea that all classifiers perform equally well, and the Bonferroni-Dum post-hoc test indicates that $\widehat{\beta}^{\theta}_{S=2}$ provides statistically significant better class probability estimates than $J48$.

### Experimental Conclusions

We have shown that a simple classifier inducer such as ID3, but with a Bayesian split metric, can reach the performance of $J48$ with a post-pruning process in terms of accuracy and class probability estimates.

Moreover, the inclusion of the Bayesian smoothing approach involved an improvement in the performance of the induced trees, but there was no statistically significant difference when the Friedman test was employed. However, the final inclusion of a heuristic to define non-uniform priors enabled this Bayesian approach to obtain class probability estimates that were better than the ones obtained by $J48$ and were statistically significant,

## 4.4 A Bayesian random split for building ensembles of classification trees

Random forest models [26] consist of an ensemble of randomized decision trees. It is one of the most outperforming classification models. With this idea in mind, in this section we introduced a random split operator based on a Bayesian approach for building a random forest. The convenience of this split method for constructing ensembles of classification trees is justified with an error bias-variance decomposition analysis. This new split operator does not clearly depend on a parameter $M$ as its random forest's counterpart, and performs better with a lower number of trees.

### 4.4.1 Introduction

The idea of randomized decision trees was first proposed two decades ago by Minger [129], but it was since ensembles of classifiers were introduced that the combination of randomized decision trees arose as a very powerful approach for supervised classification models [23; 26; 48].

Bagging [23] was one of the first approaches that exploited this idea. A group of decision trees was built over a bootstrapped replicate of the former training dataset. Finally, the last prediction was made by a majority voting criterion over the set of predictions of each single decision tree. As each decision tree was built following the usual approach [27] from different bootstrapped training data, each tree comprised a different set of split nodes. Thus, the randomization was caused by the different random variations of the bootstrapped training sets.

Another trend appeared with the use of random split node selection. For example, Dietrich et al. [48] built ensembles of trees in which at each node, the split was randomly selected from among the $M$ best splits attributes. Some years later, Breiman proposed the random forest model [26] as a combination of the bagging approach with a random split node selection. In this method, each decision tree was once again built over a bootstrapped replicate of the former training set. But, as opposed to the Diettrich et al. approach [48], first $M$ nodes were randomly selected and the best one of these was chosen. The Random Forests

outperformed the Bagging and Diettrich approaches [26]. One issue relating to Random Forests is their sensitivity to the selection of the $M$ value [26; 72], although Breiman suggested a $M$ value around the logarithm of the number of variables as a default choice.

One the main questions relating to ensembles of trees was a theoretical justification of their excellent performance. The notion of bias-variance decomposition of the error [25; 110] appeared to provide some insights. Bias represents the systematic component of the error resulting from the incapacity of the predictor to model the underlying distribution. However, variance represents the component of the error that stems from the particularities of the training sample. As both are added to the error, a bias-variance trade-off therefore takes place [110]. When we attempt to reduce bias by creating more complex models that fit better the underlying distribution of the data, we take the risk of increasing the variance component due to overfitting of the learning data. As decision trees can easily encode complex data distributions, their main disadvantage could lie in the high variance they are associated with.

Post-pruning techniques [27; 149] have been discovered as successful techniques for reducing the variance of the trees without further deterioration of their bias component [73]. With the same idea in mind, it has been seen that the success of combining multiple models relies on an underlying reduction of bias and, particularly, on the variance error component [26; 72]. A special role is played in this sense by the selection of the $M$ value in Random Forests. Higher $M$ values seem to imply low bias but higher variance and, on the other hand, lower $M$ values appear to present poorer bias but better variance [72].

In this section, we propose a new random split method derived from a Bayesian approach for building ensembles of trees. This random split is similar to the random forests one, but does not pose the problem of choosing an optimal $M$ value and allows better performance to be obtained with a lower number of trees.

The rest of the section is divided as follows. First, Section 4.4.2 attempts to justify the convenience of a random based split criterion when several classification tree models are combined in an ensemble. Section 4.4.3 introduces the Bayesian random split approach. Finally, Section 4.4.4 shows the results of the experimental evaluation.

### 4.4.2 Comparison with the random forest model

The greedy approach to building decision trees was first exposed in Section 2.3.1 and its counterpart from a Bayesian point of view in Sections 4.2.1 and 4.2.2. What we attempt to do here is to show that these greedy methods are unsuitable when the classification problem is addressed with a multiple model framework, as was described in Section 4.2.3.

As we pointed out in Section 4.2.2, at a given point of the growing process of a classification tree in a node $S_0$, there is a set of possible split attributes $(S_1, ..., S_p)$. Let us denote as $T_{S_0}$ the tree without branching at $S_0$ and $T_{S_i}$ the new tree with the inclusion of the split attribute $S_i$. For each possible tree $\{T_{S_1}, ..., T_{S_p}\}$ a score is computed, $Score(T_{S_i}) = logP(T_{S_i}|\hat{c}, \hat{x})$, using Equation (4.2).

In the deterministic greedy approach, the split node $S^*$ with the highest positive difference, $Score(T_{S_i}) - Score(T_{S_0}) > 0$, is selected. Whereas if none of the candidate split attributes shows a positive difference, the tree ceases to grow at $S_0$.

This search method appears to be highly suitable when only one classification tree is inferred from the learning data, but if one seeks a broader set of trees with high posterior probabilities so that they become selected in an approximate proportion to this posterior, the greedy approach does not appear to be very suitable.

The greedy approach is known to be very sensitive to the selection of the root node of the tree [3]. Thus, if there is a very high informative node, greedy approaches such as Bagging will probably start most of the trees of the ensembles with the same root node. Therefore, greedy search schemes mostly seem to reveal a narrow set of local maxima of the global posterior probability distribution over the different decision trees.

With this in mind, we chose a random split criterion similar to the one used in random forests. As the random selection of the split nodes at the beginning of tree appears to be more suitable than a greedy scheme, the approach presented differs from the random split of random forests in the introduction of a random condition for stopping the branching.

Information-based scores used to grow random ensembles [26] such as information gain [150] or Gini index [27] predict better partitions whenever a new split node is added. Therefore, stop criteria usually include conditions such as a minimum threshold for the number of samples or a pure partition of the data. Excessive branching implies a higher risk of over-fitting, and post-pruning techniques were therefore applied as suitable stop criteria (they reduce the size of three defining shorter rules and, in consequence, establish better stop levels).

The use of a Bayesian approach enables us to tackle the stop branching problem in an elegant manner, because of the inherent penalty they impose upon more complex models. In the previous Section 4.3, the Bayesian smoothing approach can also be seen as a Bayesian approach to tackling the stop branching problem, this combining different classification rules. In this case significant performance improvements were noted. For these reasons, possibly stopping the branching, as an additional option to be considered, appear to be justified.

### 4.4.3 A Bayesian Random Split

In this section, we present the new approach for building an ensemble of classification trees. This approach is similar to Bayesian model averaging (Equation (4.4)) which attempts to collect trees with high posterior probabilities. But the predictions of these trees are not weighted. Rather, greater importance is given to the most probable trees which appear more frequently in the ensemble of trees. Thus, this approach should be viewed as a Monte-Carlo inspired one.

As in random forests [26] (for details, see Section 2.3.2), at any node $S_0$ of the tree, $M$ split attributes $(S_1, ..., S_M)$ are randomly selected from the set of all possible split candidates. Therefore a score, $Score(T_{S_i})$, is computed for each split node $S_i$. Simultaneously, we also compute the score of the model without further splitting at this point, $Score(T_{S_0})$. Exponentiating and normalizing this vector of scores, we obtain a distribution $\Lambda_M = (\lambda_0, ..., \lambda_M)$ where each $\lambda_i$ informs us of the degree of probability of the tree model with the split node $S_i$ with respect to the rest of split candidate nodes and the tree without further splits, $T_{S_0}$. It must be remembered that $\Lambda$ is a proper probability distribution, because each

$Score(T_{S_i})$ comes from a probability itself. This would not be so evident if the scores were based on information theoretic criteria.

As $Score(T_{S_i})$ is computed with a logarithmic transformation in order to avoid overflows, normalization has to be performed as follows:

$$\lambda_i = \frac{\varphi(T_{S_i})}{\sum_{j=0}^{M} \varphi(T_{S_j})}, i \in \{0, ..., M\}$$

where $\varphi(T_{S_i})$ is scaled by the maximum score of the candidate models, $Score(T_{S_{max}})$:

$$\varphi(T_{S_i}) = e^{(Score(T_{S_i}) - Score(T_{S_{max}}))}, i \in \{0, ..., M\}$$

Finally, our approach randomly samples the split node among the $M$ candidates according to $\Lambda_M$ distribution. If the $T_{S_0}$ tree is sampled (i.e., branching is stopped at this leaf), the current $M$ split attributes are discarded and other different $M$ split attributes are randomly selected. The whole process of computing the $\Lambda_M$ is conducted again. Thus, branching stops when $T_{S_0}$ is selected and there are no more split attributes to repeat the whole process again. It is important to remark that the discarded attributes in this process can be considered again in the selection of another split node.

We now provide the pseudo-code of our Bayesian approach to a random split criterion.

**Algorithm 8** *Bayesian Random Split*

***SelectSplit***$(S_0, \vec{X} = \{X_1, ..., X_n\})$

$\vec{Z} = AvailableAttributes(S_0, \vec{X})$;

*end = false;*

*while (not end)*

$\{S_1, ..., S_M\}$ = *Random Selection*$(\vec{Z})$;

$\{S_0, S_1, ..., S_M\} = \{S_0\} \cup \{S_1, ..., S_M\}$;

$\Lambda_M = (\lambda_0, ..., \lambda_M)$=*ComputeScores*$(\{S_0, ..., S_M\})$;

$S^*$=*Sampling*$(\Lambda_M)$;

$\vec{Z} = \vec{Z} \setminus \{S_1, ..., S_M\}$;

*if* $S^* \neq S_0$ *OR* $\vec{Z} \neq \emptyset$

*end=true;*

*else if* $S^* = S_0$ *AND* $\vec{Z} \neq \emptyset$

*end=false;*

*return* $S^*$;

The function $AvailableAttributes(S_0, \vec{X})$ returns the attributes not included as split nodes in the path from $S_0$ to the root node. That is, all possible attributes available to be used as split nodes.

Random forests perform the same steps but use an information-based score instead of a Bayesian one (for details, see Section 2.3.2); they select the split node with the highest score among the $M$ candidates rather than a random sampling of the split node; and they stop branching when this reaches a pure partition or there are few samples in the partition.

### 4.4.4 Experimental Evaluation

In this section, we present the experimental results of the comparison of the Bayesian approach for random splits with the random forest one. In the first subsection we will detail the experimental approach employed and the evaluation methodology, and the second subsection presents results and conclusions. The approach presented will be denoted as Bayesian random split (BRS) as opposed to Random Forests (RF).

#### Experimental and Evaluation Setup

For these experiments we selected a set of 23 different datasets taken from the UCI repository. In Table 4.5, the datasets with their basic features are listed. In the last row, we present the range of each feature of the data sets in order to show the heterogeneity of this benchmark.

The approach presented, an ensemble of classification trees induced with a Bayesian random split, was implemented in Elvira environment [39], whereas

Table 4.5: Data Bases Description

| Name | $t$ | $n$ | $k$ |
|---|---|---|---|
| anneal | 898 | 39 | 6 |
| audiology | 226 | 70 | 24 |
| autos | 205 | 26 | 7 |
| breast-cancer | 286 | 10 | 2 |
| horse-colic | 368 | 23 | 2 |
| german-credit | 1000 | 21 | 2 |
| pima-diabetes | 768 | 9 | 2 |
| glass2 | 163 | 10 | 2 |
| hepatitis | 155 | 20 | 2 |
| hypothyroid | 3772 | 30 | 4 |
| ionosphere | 351 | 35 | 2 |
| kr-vs-kp | 3196 | 37 | 2 |
| ..... | | | |

| Name | $t$ | $n$ | $k$ |
|---|---|---|---|
| ..... | | | |
| labor | 57 | 17 | 2 |
| lymphography | 148 | 19 | 4 |
| segment | 2310 | 20 | 7 |
| sick | 3772 | 30 | 2 |
| solar-flare | 323 | 13 | 2 |
| sonar | 208 | 61 | 2 |
| soybean | 683 | 36 | 19 |
| sponge | 76 | 45 | 3 |
| vote | 435 | 17 | 2 |
| vowel | 990 | 12 | 11 |
| zoo | 101 | 17 | 7 |
| Range | 57-4k | 9-70 | 2-24 |

the experiments, along with the rest of the classifiers evaluated, were carried out in Weka platform [185]. We used non-informative Dirichlet priors over the parameters, setting the $\alpha_i$ parameters of this distribution at $1/K$ (for details see Section 4.2.1).

The data were preprocessed with the Weka filters themselves: missing values were replaced (with the mean value for continuous attributes and with the mode for the discrete ones) and discretized with the Fayyad and Irani method [58].

We evaluated the performance of the classifiers with the error rate and with a bias-variance decomposition of this error. For that purpose, we used the Weka utility, following the bias-variance decomposition of the error proposed by Kohavi and Wolpert [110] and using the experimental methodology proposed in [177].

Comparison of those performance measures followed the methodology proposed by Demsar [47] for the comparison of several classifiers over several datasets. In this methodology, the non-parametric Friedman test was used to evaluate the rejection of the hypothesis that all the classifiers perform equally well for a given significant $\alpha$ level (5% in this case). When the Friedman test detected significant differences, a post-hoc test was also used to assess particular differences among these classifiers: the Bonferroni-Dum test [47] with a 5% significance level establishing a given classifier (marked with $\star$ in the tables) as the reference model.

As in the experimental evaluation of the previous section, the ranking scores that the Friedman test assigned to each classifier (ranking scores close to 1 indicate better performance for those classifiers) were also displayed.

Both Bayesian random split and random forests were evaluated with different $M$ values and number of trees in the ensembles. Concretely, $M$ was fixed to 1, 3, 5 and equal to the logarithm of the number of variables as Breiman recommended. Four different numbers of trees were evaluated: 10, 50, 100 and 200.

In this section only a summary of the Results is presented in the tables. Fully expanded tables with the error, bias and variance for the above evaluated ensembles can be found in the Appendix: Tables 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17 and 18.

### The Role of $M$ and the number of trees in BRS

The aim of this initial analysis is to show that the Bayesian random split quickly reaches a competitive performance level with a lower number of trees and that this performance does not depend much on the $M$ value as in the case of Random Forests.

Firstly, the following comparison was made. An ensemble with the Bayesian random split was built setting the number of trees at 10 and the $M$ value at 1. This ensemble was then compared with random forest ensembles with different numbers of trees and different $M$ values.

Table 4.6 contains the different average classification errors across the 23 different databases for the different ensembles of this initial analysis. At the same time, in Table 4.7 we show the Friedman test results as applied to the different number of trees: we display the ranking scores of each ensemble along with the acceptation or rejection of the null hypothesis (all classifiers perform equally well).

As can be seen in Table 4.7, a BRS ensemble with 10 trees and $M = 1$ proves difficult to beat using Random Forest ensembles with a higher number of trees and different $M$ values, but what it is most important, there is no clear trend and Random Forests appear to beat BRS, depending on the concrete $M$ value and with a concrete number of trees.

In a second step, we exchanged the roles and tested a Random Forest ensemble with 10 trees and $M = LogN$ (the recommended value by [26]) against different

Table 4.6: Evaluating BRS ensembles with 10 Trees - Average Error

| | $\star BRS$ | RF | | | |
|---|---|---|---|---|---|
| RF Trees | M=1 | M=1 | M=3 | M=5 | M=Log N |
| 10 | 0.147 | 0.169 | 0.161 | 0.159 | 0.159 |
| 50 | 0.147 | 0.146 | 0.142 | 0.144 | 0.143 |
| 100 | 0.147 | 0.143 | 0.139 | 0.141 | 0.140 |
| 200 | 0.147 | 0.141 | 0.138 | 0.140 | 0.139 |

Table 4.7: Evaluating BRS ensembles with 10 Trees - Ranking Scores

| | $\star BRS$ | RF | | | | |
|---|---|---|---|---|---|---|
| RF Trees | M=1 | M=1 | M=3 | M=5 | M=Log N | Friedman Test |
| 10 | 2.0 | $3.9^{\perp}$ | $3.1^{\perp}$ | 2.9 | 3.1 | Reject |
| 50 | 3.4 | 3.5 | 2.6 | 2.9 | 2.7 | Accept |
| 100 | 3.7 | 3.2 | $2.3^{\top}$ | 2.8 | 2.9 | Reject |
| 200 | 3.8 | 3.1 | 2.7 | 2.8 | 2.6 | Accept |

$\top$, $\perp$ statistically significant improvement or degradation respect to BRS.

Table 4.8: Evaluating Random Forests with 10 Trees - Average Error

| | $\star RF$ | BRS | | | |
|---|---|---|---|---|---|
| BRS Trees | M=Log N | M=1 | M=3 | M=5 | M=Log N |
| 10 | 0.159 | 0.147 | 0.145 | 0.147 | 0.148 |
| 50 | 0.159 | 0.135 | 0.135 | 0.138 | 0.137 |
| 100 | 0.159 | 0.133 | 0.134 | 0.137 | 0.137 |
| 200 | 0.159 | 0.132 | 0.134 | 0.138 | 0.136 |

Table 4.9: Evaluating Random Forests with 10 Trees - Ranking Scores

| | $\star RF$ | BRS | | | | |
|---|---|---|---|---|---|---|
| BRS Trees | M=Log N | M=1 | M=3 | M=5 | M=Log N | Friedman Test |
| 10 | 4.3 | 3.2 | $2.1^{\top}$ | $2.6^{\top}$ | $2.8^{\top}$ | Reject |
| 50 | 4.9 | $2.7^{\top}$ | $2.3^{\top}$ | $2.5^{\top}$ | $2.6^{\top}$ | Reject |
| 100 | 5.0 | $2.3^{\top}$ | $2.5^{\top}$ | $2.7^{\top}$ | $2.6^{\top}$ | Reject |
| 200 | 5.0 | $2.4^{\top}$ | $2.4^{\top}$ | $2.7^{\top}$ | $2.5^{\top}$ | Reject |

$\top$, $\perp$ statistically significant improvement or degradation respect to RF.

BRS ensembles with different tree sizes and $M$ values. The analogous results are presented for this new analysis in Table 4.8 and 4.9.

In this new analysis, the trend is much clearer than in the previous case. As can be seen in Table 4.9, the BRS ensembles now robustly outperform the Random Forest ensembles with different $M$ values and different numbers of trees.

The first conclusion seems clear - Bayesian Forests reach a high performance level with a low number of trees and this performance does not depend much upon the concrete $M$ value, as in the case of Random Forests. Throughout the next subsection, we will show how this trend mainly results from a better trade-off between the bias and the variance obtained with the Bayesian random split operator.

Table 4.10: Error, Bias and Variance averaged values for ensembles with 200 trees.

| | BRS | RF | | | |
|---|---|---|---|---|---|
| | M=1 | M=1 | M=3 | M=5 | M=Log N |
| Error | 0.132[1] | 0.141 | 0.138 | 0.140 | 0.139 |
| Bias | 0.088[1] | 0.097 | 0.093 | 0.093 | 0.092 |
| Variance | 0.044[1] | 0.044[1] | 0.045 | 0.048 | 0.047 |

[1] indicates the best average error with the same number of trees.

### Bias-Variance Analysis

Herein we conducted a bias-variance decomposition of the error for both the BRS and RF models. With the aim of simplifying the result analysis, we evaluated the BRS models with $M = 1$. Analyzing the results of the previous section devoted to the role of $M$ in BRS models, we did not find any good reason to prefer a specific $M$ value. The BRS with $M = 1$ appeared to stand out somewhat more than the others.

In Table 4.10, we first give the averaged values for error, bias and variance for ensembles with 200 trees. The super-index ($^1$) indicates the best values across the different ensembles with the same number of trees. As can be seen, the BRS ensembles exhibit better averaged performance for error, bias and variance. Although not many conclusions can be extracted from this comparison, it is convenient to start by pointing out the best trade-off among bias-variance of the Bayesian random split operator. Furthermore, as was previously stated in Section 4.4.1, in random forests, higher $M$ values are associated with better a bias, but which presents a higher variance.

A more profound analysis performed out using Demsar's methodology [47]. Error (Table 4.11), Bias (Table 4.12) and Variance (Table 4.13) were compared between the Bayesian random split operator and Random Forests. We provide the ranking score of each approach and show whether the Friedman test accepted or rejected the null-hypothesis (all classifier performs equally well). The tests were performed independently for the different numbers of trees in the ensembles.

Table 4.11: Error - Ranking Scores

| | $\star BRS$ | RF | | | | |
|---|---|---|---|---|---|---|
| Trees | M=1 | M=1 | M=3 | M=5 | M=Log N | Friedman Test |
| 10 | 2.0[1] | 3.9[⊥] | 3.1 | 2.9 | 3.1 | Reject |
| 50 | 2.4[1] | 3.7 | 2.9 | 3.1 | 2.9 | Accept |
| 100 | 2.5[1] | 3.6 | 2.7 | 3.1 | 3.2 | Accept |
| 200 | 2.4[1] | 3.5 | 3.1 | 3.1 | 2.9 | Accept |

[⊥] indicates this classifier is statistically worst than the respective BRS model.

For the error, Table 4.11, only with 10 trees there are significant differences among the classifiers. In that case, Bonferroni-Dum Test [47] says that the Bayesian random split is significantly better than random forests with $M = 1$ (its ranking is marked with $^{\perp}$). For a higher number of trees, although no significant differences were found, our approach always provided the best ranking. For random forests, $M = LogN$ is seen to be the best option.

Table 4.12, shows the bias evaluation results. As was mentioned in Section 4.4.1, the random forest model with $M = 1$ presents the worst bias, which can be observed in this table. The Bonferroni-Dum test reveals significant differences of RF $M = 1$ with respect to the BRS. There is no difference with respect to the rest, but the Bayesian random split model clearly shows a better ranking across the different numbers of trees. Although the BRS exhibits a $M$ value fixed to 1, it achieves the best bias. This is a good indication, as the randomness introduction in the split criteria through a Bayesian approach indicates a promising method for further improvements.

Lastly, we evaluate the variance component (Table 4.13). In this case, the non-parametric test indicates non significant differences among the classifiers, although RF ($M = 1$) appears to stand out somewhat, with 200 trees.

## Experimental Conclusions

The value of $M$ in Random Forests has been known to affect the performance of the ensembles [26]. In a bias-variance analysis, it was shown [72] that lower $M$ values reduce variance, but increase bias and viceversa. $M = LogN$ seems

Table 4.12: Bias - Ranking Scores

| | $\star BRS$ | RF | | | | |
|---|---|---|---|---|---|---|
| Trees | M=1 | M=1 | M=3 | M=5 | M=Log N | Friedman Test |
| 10 | 2.5[1] | 3.8 | 2.9 | 3.0 | 2.9 | Accept |
| 50 | 2.2[1] | 3.8$^{\perp}$ | 3.0 | 3.1 | 2.9 | Reject |
| 100 | 2.1[1] | 3.8$^{\perp}$ | 2.8 | 3.0 | 3.2 | Reject |
| 200 | 2.3[1] | 3.9$^{\perp}$ | 3.0 | 3.0 | 2.7 | Reject |

$^{\perp}$ indicates this classifier is statistically worst than the respective BRS model.

Table 4.13: Variance - Ranking Scores

| | $\star BRS$ | RF | | | | |
|---|---|---|---|---|---|---|
| Trees | M=1 | M=1 | M=3 | M=5 | M=Log N | Friedman Test |
| 10 | 2.3[1] | 3.5 | 3.2 | 3.0 | 3.0 | Accept |
| 50 | 2.8[1] | 2.9 | 3.0 | 3.2 | 3.0 | Accept |
| 100 | 2.9 | 3.0 | 2.9 | 3.3 | 2.8[1] | Accept |
| 200 | 2.8 | 2.4[1] | 3.0 | 3.5 | 3.2 | Accept |

to present the best trade-off between bias and variance and, in consequence, the best error rate. Our experiments confirm this trend.

This trend is broken with the introduction of more randomness in the split criteria. In BRS ensembles with $M = 1$, the low variance is maintained, while the bias shows a noteworthy decrease. Thus, we achieve the best trade-off between bias and variance. Although we did not find any significant differences between random forests with $M = LogN$, the good behaviour of our new random split provides the possibility to develop new approaches with a stronger theoretical basis.

## 4.5 Conclusions and Future Work

Throughout this Chapter 4 we introduced a Bayesian approach to the problem of inferring classification trees. Concretely, in Section 4.3, we addressed the problem of estimating class probabilities, using a smoothing approach that attempts to simulate a post-pruning process, while in Section 4.4.3, we tackled the problem of dealing with several classification tree models by building an ensemble of trees. In both cases, the application of Bayesian-inspired approaches was encouragingly positive.

In short, in Section 4.3 we present a method for inducing classification trees with a Bayesian model selection approach as a split criterion and with a Bayesian model-averaging inspired approach aimed at estimating the probability of class values. We also introduced a new approach to define non-uniform priors over the parameters of the models. In order to show the good performance of this approach, we made an experimental evaluation using 27 different UCI datasets, comparing it with one of the state-of-the-art tree inducers, $J48$.

Moreover, in Section 4.4.3 we presented a new random split operator for building ensembles of classification trees based on Bayesian ideas. We also depicted the method for constructing ensembles of classification trees using this random split through a Bayesian approach. In an experimental study, we showed that this new split operator does not clearly depend upon the $M$ parameter, like its counterpart of the random forests models, and performs better with a lower number of trees. These advantages were justified with the use of a bias-variance decomposition of

the error. In random forests, $M = LogN$ attempts to find a balance between bias and variance. With the Bayesian random split with $M = 1$ presented, the low variance is maintained while the bias is clearly improved.

From our point of view, both studies provide some insights into how to address the building of single classification trees and ensembles thereof through a Bayesian approach, and propose new methods for dealing with these complex problems. There is a need for further experiments and, particularly, for theoretical developments.

# Part III

# Applications to Genomics

# Chapter 5

# Introduction to Supervised Classification of Gene Expression Data

## 5.1 An Overview of Gene Expression Data

In this section we provide a general introduction to gene expression data along with a general description of the methodology employed to obtain these data. Finally, we show possible applications and an introduction to the use of automatic classification procedures.

### Structural Genomics

Throughout the last decade, the progressive development of automatic methods for extracting DNA samples, as well as their sequencing and subsequent reading, has enabled several high-scale DNA sequencing projects. In 1997 the genome of the first organism, Saccharomyces cerevisiae, was described. Two years later it was the earthworm, Caenorhabditis elegans. Half way through the year 2000, it became possible to describe the genome of the fruit fly, *Drosophila melanogaster*, and at the end of that year it was published the genome of one plant, *Arabidopsis arrives.*

The international consortium, comprising 20 different groups of different countries, along with the private company *Celera Genomics* made public, on February 12th 2001, the provisional map of the human genome (HG), which provided extraordinary information on the human genetics bases. The group of this international consortium, led by Eric Land, from *Sanger Center* (Cambridge, UK), published the complete sequence in the journal *Nature*, while the American company *Celera Genomics*, led by Craig Venter, published the same sequence in the another famous journal *Science*. The international consortium estimated that the human genome contains around 31,780 genes codifying certain proteins, and until that date had discovered around 22,000 genes. *Celera* claimed to have some evidence of the existence of 26,000 genes, and also estimated that there were around 38,000 genes. Although the Human Genome Project (HGP) was completed in April 2003, the exact number of genes is as yet unknown.


Figure 5.1: *Science* and *Nature* Front Pages

The sequence obtained is enormously significant, with many interesting points:

- Humans have only twice as many genes as the fruit fly, one third more than the earthworm and only 5000 more than the plant Arabidopsis. At least 98% of human DNA is identical to the chimpanzee's and that of other primates.

- Genes are made up of 3200 millions of base pairs, shared among 23 chromosome pairs. The denser chromosomes (with more genes codifying proteins)

are numbers 17, 19 and 22. Chromosomes X, Y, 4, 18 and 23 are the most arid ones.

- The Celera equipment used samples from twelve people to sequence the human genome. Each person shares 99.99% of the genetic code with the remainder of human beings. Only 1250 letters separate one person from another.
- To date, 223 genes have been found to be similar to bacterial genes.
- Only 5% of the genome codifies proteins, while 25% of human genome is almost void, with large free spaces existing between one gene and another.
- It is estimated that there are around 250-300,000 different proteins. Therefore, each gene could be involved, on average, in the synthesis of ten proteins.
- Somewhat more than 35% of the genome contains repeated sequences, this part being known as garbage DNA.
- A high number of small variations among genes has been identified, which is known as single nucleotide polymorphism (SNP). Most of them do not have a specific clinical effect but, for example, whether a person is sensitive or not to a given drug, or prone to suffer a given disease, depends on them.

A huge amount of information needs to be analyzed, for example, there is a need to establish where genes start and finish, as well as to identify their exons, introns and regulatory sequences. There is also a need to make comparisons of sequences of several species (Comparative Genomics). The sequence map generated by this project is being used as a primary information source for human biology and medicine. The public project led by the governments of the USA and of several European countries have introduced all the information into a free-access database [1].

Now that the Genome has been decoded, the great scientific challenge involves investigating how genes interact and how the tiniest alterations in each of these interactions predisposes a person to suffer a disease. Understating how genetic

variants regulate cell phenotype, tissues and organs will constitute the objective of research in the next century. It is estimated that there are about 8000 hereditary diseases, but at present only 200 can be detected before birth.

## Functional Genomics. The post-genomic age.

Structural Genomics is the branch of Genomics dealing with characterization and localization of sequences forming the DNA of genes, thus allowing genetic maps of organisms to be created. *Functional Genomics* is a research field dealing with collection of information on the function of genes. Knowledge provided by structural Genomics is essential to achieving this. Furthermore, the experimental methodologies employed must be combined with computational analysis of the results, due to the huge volume of information generated in these studies.

The aim of *Functional Genomics* is to fill the gap between the existing knowledge of the sequences of a gene and its functionality, in order to disclose the behaviour of biological systems. Biological research into the role of single proteins and genes must be expanded to the study of all of these as a whole.

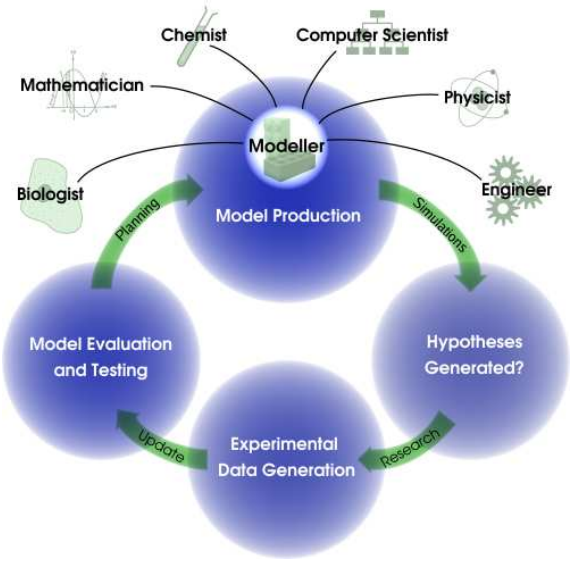


Figure 5.2: The Post Genomic Age

## Biochips

The technological bases of Biochips lie in the development and minimization of the affinity techniques that have been employed for years as common tools in molecular biology. Development of initial pilot tests of affinity with immobilized DNA over solid substrates began in the sixties with the first immunity pilot tests. The next step arrived in the following decade when Edwin Sourthern [164] began to employ filters of nitrocellulose to act as solid substrates for joining DNA molecules. The immobilized DNA did not interact with other immobilized molecules, but rather maintained its hybridization capacity with complementary dissolute molecules. Detection of these hybridizations was performed by means of detection of radioactive markers. This type of technique was called as *Southern blot* and was later extended to the field of protein and RNA immobilization.

With the refinement of the *Southern blot* technique, the next step in the path towards the emergence of Biochips consisted of developing immobilized biological material matrixes, with the use of porous surfaces such as nitrocellulose or nylon membranes.

Later, researchers started to work with surfaces with smaller pores and with solid substrates, such as silicon or glass. At the same time, with the arrival and development of miniaturization techniques, the size of these pores was reduced, with the resulting higher density in these matrixes. The whole process finally led to the development of micro-matrixes.

One of the most important events was at the end of the eighties when in an Affymax laboratory, where a group of four researchers, Stephen Fodor, Michael Pirrung, Leighton Read and Lubert Stryer, working on synthesis of polypeptides over solid substrates, developed the GeneChip platform. The relevance of this step lies in the great miniaturization capacity achieved by this system. The technology developed by Affymetrix (the new brand of Affymax) subsequently led to the rapid appearance of new companies and new developments, which brought about the current high grade of technological diversity.

## Methodology for obtaining gene expression data

The methodology for conducting an experiment with a Biochip platform is divided into two main steps, and some of these are conditioned by the type of Biochip employed in the experiment. The methodology involved is basically the following:

**Biochip Design:** During the design process, we determine the type and the quantity of biological material to be immobilized over the surface of the solid substrate. This varies depending on the type of experiment. The density of the integration is also selected.

**Biochip Manufacturing:** This step is highly diversified as a consequence of the large amount of technological solutions existing on the market. It determines the density of the integration that can be achieved in the chip. In general, the chips manufactured by the big companies provide the highest integration densities.

**Sample Preparation:** In this step, the biological sample is subjected to a set of required processes in order to prepare it for this kind of experiments. The process consists of extracting and purifying the material to be analyzed (DNA, RNA or proteins), an amplification phase and, lastly, marking the biological samples to allow their detection in the revealed process. The most common markers are fluorescent, although radioactive markers can also be used.

**Hybridization and Washing:** This step is practically the same for all commercial chips and for the personalized ones too. It is a key step because it consists of the affinity reaction in which the DNA of the marked samples is hybridized for subsequent identification. Washing is performed to remove the non-specific interactions produced by the sample and by the surface of the Biochip.

**Results Reading:** This process is conditioned by the great variety of technological solutions. Among these solutions, the most common ones are the utilization of laser scanners and CCD cameras for detecting the fluorescent markers with which the sample was marked.

**Results Storage:** After the Biochips are developed, the results must be stored in an electronic device.

**Results Analysis:** This is the final phase of the experiments based on Biochips. In this step, data are provided by the development process and these can be presented in numeric form or in a 16 bits image. In these data, one can see the points at which the hybridization reaction was positive, and when no reaction took place. At the present time, Bioinformatic analytical software is employed to extract relevant conclusions from the experiment.

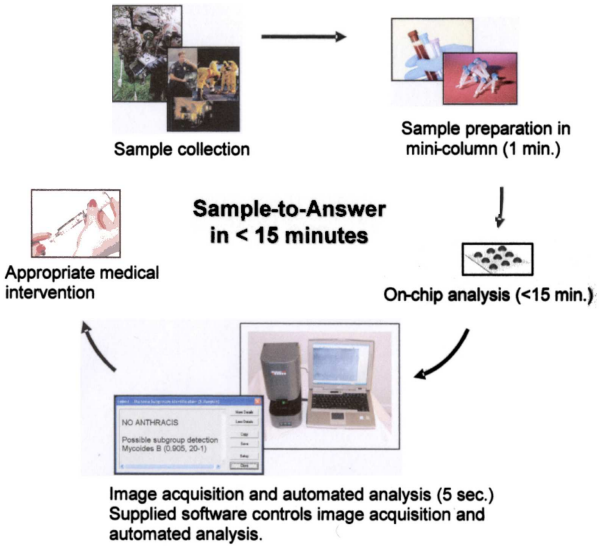


Figure 5.3: Biochips Cycle

## Applications

Technology based on Biochips is being applied to very different types of studies and applications. Some of these are:

**Genetic Supervision:** It enables the simultaneous quantification of the expression of a very high number of genes. It also allows a quantitative approach for determining expression patterns, as well as study of gene functionalities,

in order to identify genes activated in a different way when submitted to different conditions, for example, a tumoral process.

**Polymorphism and Mutation Detection:** It allows study of all possible polymorphism, as well as detection of mutations of complex genes. The meaning in the variation of human genetics is analyzed through a correlation of the mutations of normal gene sequences with respect to those of specific diseases.

**Clinical Diagnostic:** Biochips are used in microbiology with many objectives in mind: biological comprehension of microorganisms, development of preventive measures against mortal diseases. Furthermore, microarrays can also be employed in the analysis of clinical aspects for diagnosis of different kinds of tumours.

**Screening and Drug toxicology:** The idea involves analyzing the rapid transformations of genetic expression profiles that take place when a patient is administered a drug.

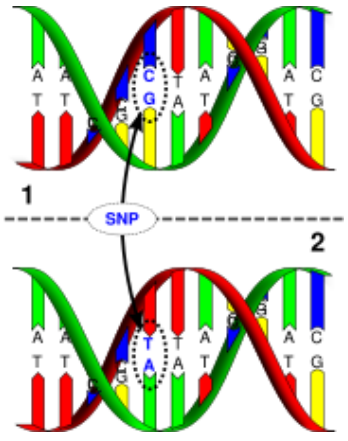


Figure 5.4: Single Nucleotide Polymorphism

## 5.2 An Introduction to Supervised Classification of Gene Expression Data

As we pointed out in the previous section, the study of the expressions of thousand of genes in a single clinical test enables us to perform comparisons among tissues, pathological phases or the different responses to different biological conditions. Consequently, a huge amount of data is generated, which needs to be assessed and analyzed.

This technology has been successfully applied to a wide range of carcinogenic diseases such as breast cancer [173], embryonic central nervous system cancer [144], colon cancer [7], Hopkins lymphoma [5], etc. The main contributions of these studies involve the description of new disease subtypes that were indistinguishable under the current diagnosis methods, and they have raised important open questions to the research community, due to the heterogeneity of the responses to many cancer treatments. Early diagnosis for some kinds of cancer diseases, such as ovarian cancer or colon cancer, was a big problem in medical research prior to the emergence of these tools. One of the main drawbacks of this approach lies in the identification of genes directly involved in the biological processes leading to these pathologies. This problem arises because, in the experiments, thousands of genes are analyzed, while only a few are usually relevant.

Supervised classification techniques (Chapter 2) have been applied to the analysis of genomic data (see [66] for a review of these applications). Apart from supervised classification, another two important research lines have been successfully exploited:

**Gene Regulatory Networks or Genetic Networks:** They are Network-based representations that attempt to encode gene relations. Many studies have been published with the aim of inferring gene regulatory Networks. [69; 70; 74; 79; 88; 139; 167; 169; 191]

**Dynamic Bayesian Networks:** Gene Expressions should be considered to involve a temporal process that varies throughout the cellular cycle, and these networks are therefore intended to achieve better modelling of these expressions [13; 86; 107; 133; 143].

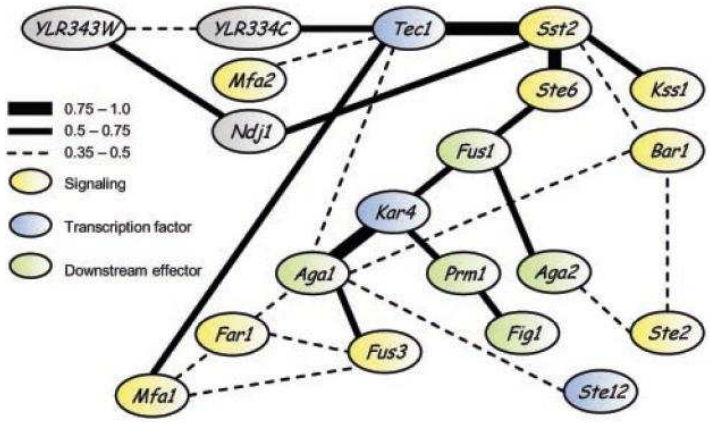


Figure 5.5: Gene Regulatory Networks or Genetic Networks

Bayesian Classifiers represent a very active line of research in the field of gene expression data. They are often combined with gene selection methods to improve classification performance by removing noisy and irrelevant genes but, in particular, to determine which genes really define and predict the class values and, therefore, are involved in the underlying biological process leading to the different values of the variable of interest.

For example, in [12] several gene selection methods were presented. Each one was evaluated using a Naive Bayes. Moler et al. [131] also used the Naive Bayes and the Support Vector Machine model, another competitive classifier, for gene selection. Several learning methods (IB1, NB, C4.5 and CN2) were employed by [94] for cancer prediction. This study thereof focuses upon a gene selection process which uses a wrapper approach. In a posterior extension of the previous study, [93], it was jointly applied filter and wrapper techniques for the gene selection task. They applied two types of filter measures: continuous (p-metric and t-metric) and discretized (Shannon's entropy, Euclidean distance, Kullback-Leibler divergence). They validated the results once again with IB1, NB, C4.5 and CN2. Better results were obtained when data were discretized. Many other studies apply supervised classification to gene expression data problems [16; 17; 35; 65; 81; 87; 122; 140; 192]

## 5.3 An Overview of Diffuse Large-B-Cell Lymphoma

### Lymphoma:

Lymphoma is a cancer of the white blood cells, namely lymphocytes, which forms the lymphatic system. There are two main types of lymphoma: Hodgkin lymphoma and non-Hodgkin lymphoma. Lymphoma represents the most common blood cancer and the third most common cancer in children. Lymphoma occurs when lymphocytes, a type of white blood cell, present abnormal growth. The body has two types of lymphocytes: B lymphocytes, or B-cells, and T lymphocytes, or T-cells. B-cell lymphomas are more common, developing into lymphomas, although both cell types can develop it. As with normal lymphocytes, those that turn malignant can grow in many parts of the body, including the lymph nodes, spleen, bone marrow, blood or other organs.

### Non-Hodgkin lymphoma:

There are around 35 types of lymphoma, 30 of these being classified as non-Hodgkin lymphoma (NHL). Nearly all non-Hodgkin lymphoma cases occur in adults, with average age of diagnosis in the 60s. Scientists do not yet know the exact causes of non-Hodgkin lymphoma. Most people diagnosed with non-Hodgkin lymphoma do not belong to a risk group, although an increasing number of scientists believe that infections may play an important role in causing some types of non-Hodgkin lymphoma.

### Diffuse large B-cell lymphoma:

Diffuse large B-cell lymphoma (DLBLC) is the most common of the non-Hodgkin lymphomas, accounting for up to 30 percent of newly diagnosed cases. Diffuse large B-cell lymphoma is an aggressive, or fast-growing lymphoma. It can arise in lymph nodes or outside the lymphatic system, in the gastrointestinal tract, testes, thyroid, skin, breast, bone or brain. Often, the first sign of diffuse large B-cell lymphoma is painless or occasionally painful and presents rapid swelling

in the neck, armpit or groin caused by enlarged lymph nodes. Other symptoms include night-time sweating, unexplained fevers and weight loss.

## Diagnosis of Diffuse large B-cell lymphoma:

Doctors usually diagnose diffuse large B-cell lymphoma by taking a small sample (known as a biopsy) of the tumour and observing the cells under a microscope. They will also examine other organs, such as the spleen, liver and bone marrow. Additional tests, such as blood tests, X-rays, and scans may be used and can also help to determine how far the cancer has spread, thus indicating its stage. In stage I, the lymphoma appears only in one group of lymph nodes in a particular body region, while in patients with stage II, the disease is present in more than one lymph node group, but limited to one side of the diaphragm (midline of chest and abdomen). In contrast, patients with stage III disease have the lymphoma on both sides of the diaphragm, while those with stage IV disease have involvement of other non -lymph node organs such as the liver or bone marrow. Most patients with diffuse large B-cell lymphoma are adults, although this lymphoma is sometimes seen in children.

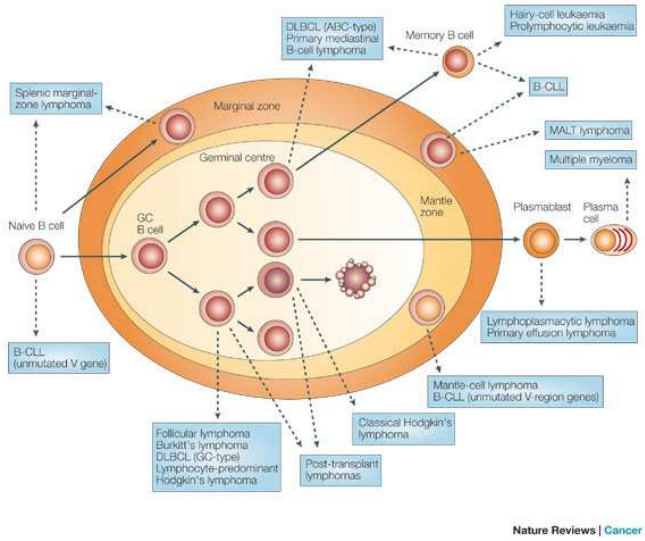


Figure 5.6: Mechanisms of B-cell lymphoma pathogenesis [115].

## Molecular Subtypes of Diffuse Large B-cell lymphoma:

One of the main problems of DLBCL is that it is clinically heterogeneous: around 40% of patients respond well to therapy and have prolonged survival, whereas the remainder succumb to the disease. Alizadeh et al. [5] found in 2000 that there was an underlying molecular heterogeneity in these tumours. They used DNA microarrays to conduct a systematic characterization of gene expression in B-cell malignancies and found that there was diversity in gene expression among tumours in DLBCL patients (Figure 5.3 shows an image of this microarray). They also found that this diversity was apparently correlated with the variation in tumour proliferation rate, host response and differentiation state of the tumour. They therefore conclude that there are actually two distinct molecular forms of DLBCL:

**Activated B Cell-like (ABC):** with a pattern of genetic expression that is similar to healthy, activated B cells.

**Germinal center B Cell-like (GCB):** with a pattern of genetic expression that is similar to germinal center B cells and a chromosomal translocation involving the gene bcl-2.

There were some remaining DLBCL cases, called "Type III", that are unrelated to any of the two subclasses, but that are used as a control group.

Most important, patients with germinal center B-like DLBCL had a significantly higher overall survival rate than those with activated B-like DLBCL.

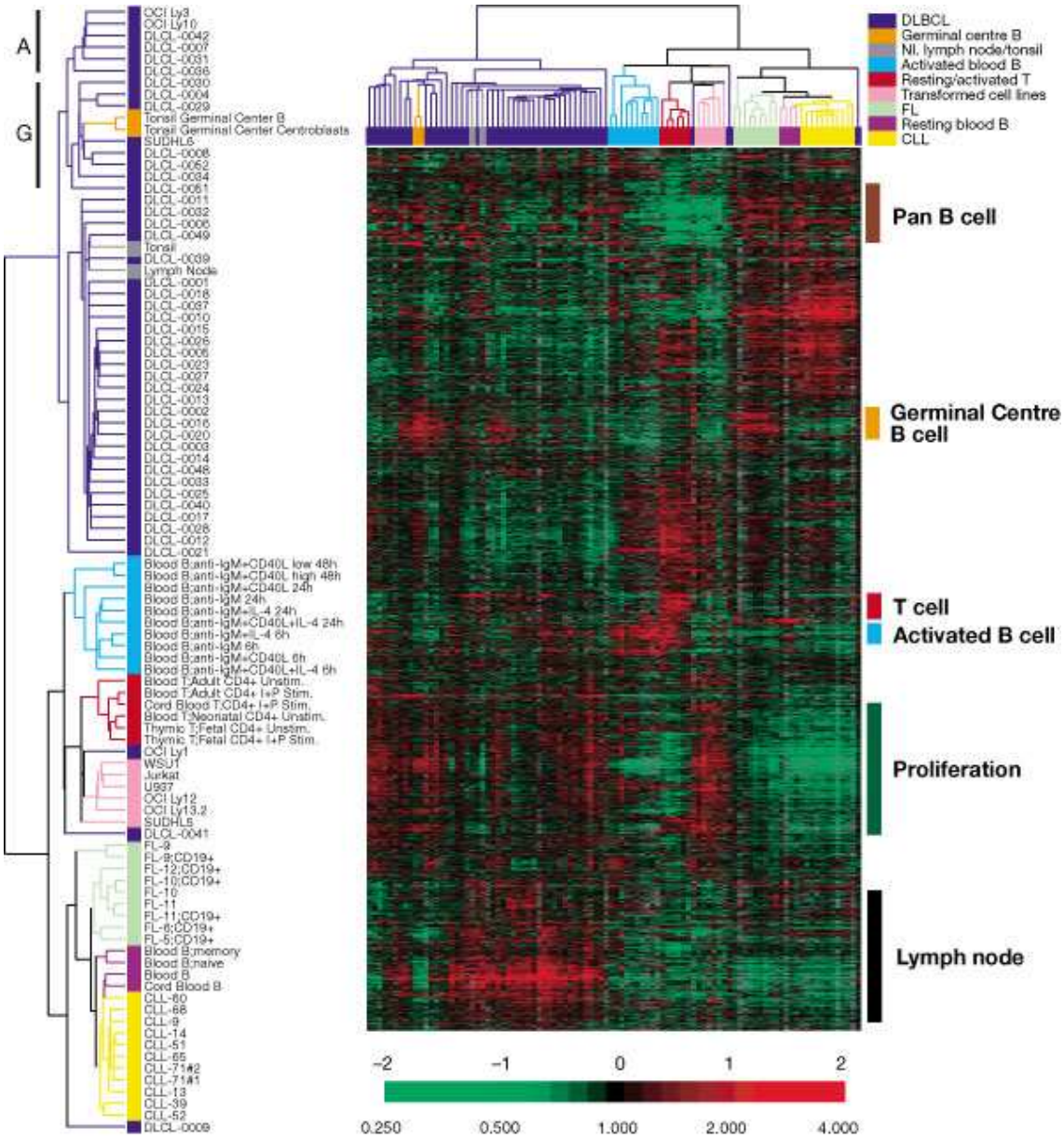


Figure 5.7: Image of microarray results obtained by Alizadeh et al. [5].

# Chapter 6

# Selective Gaussian Naive Bayes Models for DLBCL Classification

In this chapter we present two new versions of a Selective Naive Bayes classification model that deal with the peculiarities of gene expression data. These models perform a gene selection for the the problem of classifying Diffuse Large B-Cell Lymphoma (DLBCL) in two subtypes: *Activated B Cell-like* (ABC) and *Germinal Center B Cell-like* (GCB) (see Chapter 5 for an introduction to this problem).

## 6.1 Motivation

DLBCL gene expression data, like other gene expression datasets, have several characteristics preventing direct adaptation of standard classification models. From an automatic learning point of view, the following are the three main features requiring the design of classification models that consider these specific issues:

**High Dimensionality:** In Section 5 it was shown how Biochips can analyze thousand of genes in one single experiment. Thus, genomics research exploits this possibility and attempts to analyze as many genes as possible. Therefore, gene expression datasets usually involve several thousands of

variables. In the case of DLBCL, the datasets employed have around 8000 genes.

**Reduced Size:** Although Biochips can evaluate thousand of genes effortlessly, a different issue involves obtaining a high number of samples from which to extract their genetic profiles. As was shown in Section 5.1, the process for preparing biological tissue for analysis is a complex one and can not be performed in a totally automatic manner. Furthermore, the availability of a high number of patients is also another problematic issue. It is therefore quite common to have gene expression datasets with a few tens of samples. Concretely, the DLBCL dataset has a little over 200.

**Very Noisy:** Noise is another component to be taken into account on analysing this kind of data. Processes such as gene marking, hybridization reaction and extraction of results with CCD cameras to which gene analysis is submitted, introduce an important noisy component in the continuous values representing their activity index.

For the first issue, the high dimensionality, *Feature Selection* (Section 2.4) appears to constitute a suitable approximation that can reduce the number of variables detecting irrelevant genes. Although wrapper methods (Section 2.4.2) are the most powerful in the general case, their direct application is not feasible due to the high number of variables. Thus, the combined used of a quick filter approach (Section 2.4.1) with a wrapper feature selection method seems to be a reasonable option and will therefore be the one used in this study.

The small size of gene expression datasets obliges us to avoid very complex classification models demanding a high number of parameters. With a sample size of around a few hundred samples and a strong noisy component, parameter estimations become less accurate and the final classification model can suffer from over-fitting. In the approaches presented herein, a Gaussian Naive Bayes model ( Section 2.2.1) is employed because, as these models have proved to be very competitive in a broad range of classification problems (Section 2.2.1) and their simplicity is especially effective in a domain in which the number of samples is scarce.

Another issue is how to handle continuous data. Previous work with Bayesian classifiers has solved this problem by discretizating them [84] or assuming that the predictive variables have a Gaussian distribution [42; 99]. The latter option is taken in these approaches because Gaussian distributions present a good trade-off between approximation capacity and reduced number of parameters (only the mean and deviation have to be estimated). Concretely, independence and normality of the variables shall be assumed when the class value is known.

## 6.2 The Selective Gaussian Naive-Bayes Model

The models we propose in this chapter are based on the Selective Naive Bayes model [119], previously introduced in Section 2.2.2. In this section this model is briefly described, the specific notation being given in context for this problem.

### Gaussian naive Bayes classifier

We shall use $\mathbf{G} = \{G_1, \ldots, G_n\}$ to denote the set of genes describing the possible samples to be classified ($G_i$ is the variable related with the $i$-th gene), and $C$ is the class variable with two classes (ABC and GCB) corresponding to the two subtypes of DLBCL. The classification problem reduces to find $c^*$ such as:

$$c^* = arg_c \max P(C = c | G_1 = g_1, \ldots, G_n = g_n)$$

Let us denote $D = \{\vec{c}, \vec{g}\}$ as the DLBCL data learning set with $T$ labelled instances $(c_j, \mathbf{g}_j)$ with $j = 1, ..., T$. And $\vec{g}_{i|c_j}$ shall denote the projection of $D$ over the variable $G_i$ for those instances belonging to class $c_j$.

In the Naive Bayes classifier [54], no structure learning is required. It is assumed that genes $\mathbf{G} = \{G_1, \ldots, G_n\}$ are independent and distributed as a Gaussian density when the variable to classify $C$ is known. Thus, the subsequent probability of the class $c_j$ given a test case $\mathbf{g} = \{g_1, \ldots, g_n\}$ is computed as follows:

$$P(C = c_j | \mathbf{g}) \propto p(c_j) \cdot \prod_{i=1}^{n} f_{\mathrm{N}}(G_i = g_i : \mu_{ij}, \sigma_{ij})$$

where $\mu_{ij}$ is the mean and $\sigma_{ij}$ is the standard deviation of the values of the vector $\vec{g}_{i|c_j}$. And $f_{\mathcal{N}}$ is the density function of a Gaussian distribution (see Section 2.2.1 for details).

### Wrapper feature selection

Wrapper Feature Selection (WFS) begins with an empty set of selected genes, and successively adds the gene $G_{max} \in \mathbf{G}$ that maximizes a given *evaluation function*. This is known in the literature as *Forward Sequential Selection* (FSS) [118]. We use the accuracy of the classification as the evaluation function. This score is obtained by the application of a Gaussian Naive Bayes classifier using a *leave-one-out cross-validation* (LOO) scheme [168].

Let $\mathbf{F}_l$ be the set of selected features in step $l$ of the WFS algorithm. Then, in step $l+1$, a new Gaussian Naive Bayes model is learned with the set of features $\mathbf{F}_{l+1} = \mathbf{F}_l \cup \{G_{max}\}$ being $G_{\max}$, the gene that maximizes the increment in classification accuracy in the training data set $D$ using LOO validation. The WFS algorithm continues selecting new features until a given *stop criterion* is verified. Suppose $Acc(\mathbf{F}_l)$ is the classification accuracy in step $l$ with the set of features $\mathbf{F}_l$. The algorithm stops if $Max\{Acc(\mathbf{F}_l), Acc(\mathbf{F}_{l-1}), \ldots, Acc(\mathbf{F}_{l-q+1})\} \leq Acc(\mathbf{F}_{l-q})$, where $q$ is a given parameter of the algorithm. That is, the algorithm stops when $q$ consecutive steps are performed without an improvement in the classification accuracy.

## 6.3 A Filter-Wrapper Approach with an Abduction Phase

In this section we present a new type of Selective Naive Bayes classification model that handles continuous data and makes a gene subset selection in two stages.

As there is a large number of genes and it is unfeasible to use wrapper selection over all of them, we designed a two-step procedure. The first step is based on ANOVA (a filter measure, Section 2.4.1) that performs a one-way analysis of variance for each gene, in an attempt to obtain the most relevant and not correlated genes. The second gene selection method is a search method comprising

two substeps: a wrapper phase and the *abduction* phase. The wrapper phase selects a set of genes using the classification accuracy as the evaluation function. One problem is that, due to the small number of instances in microarrays, the selected genes have an important random component: small variations in the training data can produce very different sets of genes. To increase the robustness of the wrapper phase, an abduction phase is subsequently applied, which consists of repeating the selection with different partitions of the training data and then learning a Bayesian network that attempts to discover the patterns in the different runs of the wrappers phase. By applying an abduction algorithm to the learned Bayesian network, we can obtain the *K-most probable configurations* of the variables of the net. These configurations shall correspond to the most likely genes to be selected by the wrapper method.

This proposal is validated to select relevant genes for the classification of instances of Diffuse Large B-Cell Lymphoma in two classes: the germinal centre B cell-like (GCB) group and the activated B cell-like (ABC) group.

The rest of the section is organized as follows, Section 6.3.2 describes the details of our proposed method. Section 6.3.3 gives the details of the experimental validation (the parameters used in the experimental setup, the experimental results and a comparison with the results of [125; 189] for the same problem).

### 6.3.1 Filter Anova phase

In this phase, we performed a one-way analysis of variance for each gene of the dataset in order to select a subset $\mathbf{G}_A \subseteq \mathbf{G}$ with the most relevant genes (a similar approach was employed in [53]). The idea is to select genes $G_i$ with a significant difference between their means for the subclasses and not correlated with other genes $G_j$. For this purpose, we used the F statistic. This statistic is used in the literature to establish whether the means of a finite set of populations are the same.

Given a gene $G_i$, the $F$ statistic tests the hypothesis that the means of the two set of values $\vec{g}_{i|c_1}$ and $\vec{g}_{i|c_2}$ are the same. If the hypothesis is accepted, then $G_i$ is not a good candidate to be included as a feature of the classifier. When there is a big difference between the two subgroups, then the value $F$ will be high, too.

In order to remove redundant genes, this approach searches for the set of genes correlated with a given $G_i$ considering only one class $c_j$, this set is denoted as $R_j(G_i)$, $(j = 1, 2)$. To obtain $R_j(G_i)$, the Pearson correlation coefficient $\rho$ is calculated using the $\vec{g}_{i|c_j}$ vectors. The $\rho$ parameter takes values in the continuous interval $[-1, 1]$. When $\rho$ is near 1.0 or $-1.0$ then the variables are correlated.

We consider that a feature $G_l$ belongs to $R_j(G_i)$ $(j = 1, 2)$ if $\underline{\rho}(\vec{g}_{i|c_j}, \vec{g}_{l|c_j}) > \theta$, where $\underline{\rho}(,)$ is the lower limit of the confidence interval at 95% of the Pearson correlation coefficient between $G_i$ and $G_l$ and $\theta$ is a fixed threshold.

Thus, the Anova phase begins with the calculation of the $F$ statistic value for all the gene expressions. The genes are then sorted from higher to lower $F$ statistic values. Let $\mathbf{G}_s = \{G_{s(1)}, \ldots, G_{s(n)}\}$ be the sorted set of genes.

The following algorithm is applied twice to obtain two subsets $\mathbf{G}_{c_j} \subseteq \mathbf{G}$ $(j = 1, 2)$ of genes, one considering each one of the two classes:

**Algorithm 9** *Filter Anova Phase*

$\mathbf{G}_s = \{G_{s(1)}, \ldots, G_{s(n)}\}$; *//Genes sorted by F-Statistics.*

$\mathbf{G}_{c_j} = \emptyset$;

*While* $\mathbf{G}^s \neq \emptyset$

- *Include* $G_{max} \in \mathbf{G}^s$ *with highest ranking in* $\mathbf{G}_{c_j}$;
- *Calculate the set* $R_j(G_{max})$; *//Genes correlated with* $G_{max}$ *in class* $c_j$
- $\mathbf{G}^s = \mathbf{G}^s \setminus \{G_{max}\}$;
- $\mathbf{G}^s = \mathbf{G}^s \setminus R_j(G_{max})$;

*return* $\mathbf{G}_{c_j}$;

Finally, the resulting set of features in the Anova Phase is the set $\mathbf{G}_A = \mathbf{G}_{c_1} \cap \mathbf{G}_{c_2}$. That is, once a gene $G$ is selected, genes that are correlated with it in both classes $c_1$ and $c_2$ are discarded.

### 6.3.2 A wrapper method with an abduction phase

In this phase, we consider a method that combines a wrapper feature selection methodology with the application of an abduction algorithm for Bayesian networks. The wrapper selection step is run $m$ times to obtain $m$ possible sets of genes $\{\mathbf{F}_1 \dots \mathbf{F}_m\}$. The $m$ sets are used to learn a Bayesian network where each node corresponds to each one of the possible features. The idea is to select a set of genes that does not depend on the particular partition of training and test sets, by repeating the selection and then computing the set of genes with the highest probability. The started gene set is the one obtained by the above Anova phase, $\mathbf{G}_A$.

#### Wrapper feature selection

The scheme for obtaining the different sets of features is similar to the $k$-fold cross-validation [108]. It starts by decomposing the training dataset $D$ into $k$ subsets of the same number of samples $\{D_1, \dots, D_k\}$. Then, $k$ training datasets are defined as follows $T_j = D \setminus D_j$ (for each $j \in \{1, \dots, k\}$). From each $T_j$ we obtain a subset of genes $\mathbf{F}_j$.

The wrapper feature selection (WFS) algorithm (Section 6.2) is then applied $k$ times: in stage $j$ ($j \in \{1, \dots, k\}$), $T_j$ is used as the training dataset and $D_j$ as the test dataset. The Wrapper methodology begins with an empty set of selected genes, and successively adds the gene $G_{max} \in \mathbf{G}_A$ that maximizes the classification accuracy. Let $\mathbf{F}_j^l$ ($j = 1, \dots, k$) be the set of features selected in an intermediate step $l$ of the WFS algorithm. Then in step $l + 1$, a Gaussian Naive Bayes model is learned for the set of features $\mathbf{F}_j^l$ using the training dataset $T_j$. We obtain $\mathbf{F}_j^{l+1} = \mathbf{F}_j^l \cup \{G_{max}\}$ ($G_{max} \in \mathbf{G}_A$), $G_{max}$ being the gene that maximizes the increment in classification accuracy in the test dataset $D_j$. When there are several possible genes $G_{max}$, we select the one with the highest F-value. The WFS algorithm continues selecting new genes until the *stop criterion* is verified.

Assuming $W(\mathbf{F}, D)$ is the classification accuracy for the set of genes $\mathbf{F}$ in the test dataset $D$. $W(\mathbf{F}, D)$ takes values on the interval $[0.0, 1.0]$. Consider that $\Delta_l$ is equal to $\Delta_l = W(\mathbf{F}_j^{l+1}, D) - W(\mathbf{F}_j^l, D)$ (increment in classification accuracy in step $l$ by adding a new gene).

**Condition 1 (C1)**: $\Delta_l > 0.02 \cdot (l - 20)$. It attempts to avoid stopping in local minima, allowing negative increments in accuracy.

**Condition 2 (C2)**: $W(\mathbf{F}_j^l, \mathcal{D}) < 1 - l \cdot 0.001$.

Subsequently, the *stop criterion* is defined as: If Condition 1 and Condition 2 are true, then the iteration goes on; otherwise it stops. The idea of these two conditions is to allow at the beginning the addition of genes, even if accuracy is not improved, and when the number of already selected genes increases, these conditions require greater increments in performance to include new genes (**C1**), and stop when the accuracy is too high (**C2**).

### Abduction phase

The above method obtains $k$ different sets of genes $\{\mathbf{F}_1 \dots \mathbf{F}_k\}$. If the complete process is repeated $t$ times, $m = k \times t$ sets of genes shall be obtained, $\mathcal{F} = \{\mathbf{F}_1 \dots \mathbf{F}_m\}$. In this abduction phase [159] we will extract the final set of genes from $\mathcal{F}$.

Let us define a discrete variable $Y_i$ for each one of the genes $G_i$ in the set $\Phi = \cup_{j=1\dots m}\mathbf{F}_j$. Then $\mathbf{Y}$ is defined as an ordered set of discrete variables $\mathbf{Y} = \{Y_1, \dots, Y_p\}$, where $p$ is the number of features in $\Phi$. Now, let us define an instance $\mathbf{y}_j$=$(y_{1j}, \dots, y_{pj})$ $(j \in \{1, \dots, m\})$ of $\mathbf{Y}$ where $y_{ij} = 1$ if $G_i$ is included in $\mathbf{F}_j$ and $y_{ij} = 0$ if $G_i$ is not included in $\mathbf{F}_j$. A new training dataset with $m$ instances can be considered as: $M = \{\mathbf{y}_1, \dots, \mathbf{y}_m\}$.

From the data set $M$, a Bayesian network can be inferred with the K2-learning algorithm [41]. In this Bayesian network we can compute the most probable configuration $\mathbf{y}^*$ with $\mathbf{y}^* = arg\max_{\mathbf{y}} P(\mathbf{Y} = \mathbf{y})$. And by means of an abduction algorithm, we can also obtain the $K$ most probable configurations $\{y_1^*, ..., y_K^*\}$ [159]. From each configuration $\mathbf{y}_k^* = (y_{k,1}, \dots, y_{k,p})$ with $(k = 1, \dots, K)$, a candidate set of genes $\mathbf{G}_k$ is derived as follows: if $y_{k,i}^* = 1$ then $G_i$ is included into $\mathbf{G}_k$. This process gives rise to $K$ candidates sets of genes: $\{\mathbf{G}_1, \dots, \mathbf{G}_K\}$.

The final selected set of genes $\mathbf{G}^*$ is the set $\mathbf{G}_k$ that minimizes the *average log-likelihood of the true class* or *Log-Score* [153] in the complete training dataset $D$ using a Gaussian Naive Bayes classifier.

### 6.3.3 Experimental evaluation

#### Experimental setup

We evaluated this proposal with the DLBCL classification problem [5]. The dataset has been taken from [189]. This dataset contains 8503 features (clones) with 134 samples belonging to class GCB, 83 samples belonging to class ABC and 52 samples to Type III. Type III was not included in the original work of Alizadeh et al. [5] but in Wright et al. [189], this last subtype was considered in the test set to show how its elements were classified into the two main types. They argued that in a good classifier, Type III cases should be distributed fairly into classes GBC and ABC with similar frequencies.

In order to compare these results with those of [189] we applied a validation scheme in which the dataset was randomly divided into a training and test dataset. All Type III cases were included in the test set as in [189]. These sets were made up of the following sets of instances:

**Training data set** : 67 samples GCB, 42 samples ABC.

**Test data set** : 67 samples GCB, 41 samples ABC and 52 samples Type III.

This division process was repeated 10 times in order to obtain a more exact estimation of the accuracy of the model. Confidence intervals of the accuracy were also computed.

The parameters set to learn the model are the following:

- Threshold $\theta$ for the correlation coefficient in the Anova phase: $\theta = 0.15$. This value was selected because it produces around 80 genes, a feasible number for the wrapper phase. With the use of $\theta = 0.20$ we obtained 190 genes, which made the wrapper phase unfeasible in the computer in which experiments were undertaken.

- k-fold-cross validation: In the search phase (Section 6.2) we used a $k = 10$ to divide the training dataset $\mathcal{D}$ into $k$ randomly subsets. And k-fold-cross partitions were repeated 3 times, and a sample of 30 sets of selected genes was finally obtained.

Table 6.1: Mean number of cases classified in each group, using only the Anova phase.

| **Training Dataset** | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclassified |
| ABC | $40.4 \pm 1.03$ | $1.2 \pm 0.66$ | $0.4 \pm 0.37$ |
| GCB | $0.8 \pm 0.74$ | $65.9 \pm 1.42$ | $0.3 \pm 0.68$ |
| **Test Dataset** | | | |
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclassified |
| ABC | $37.3 \pm 1.57$ | $2.2 \pm 0.87$ | $1.5 \pm 0.7$ |
| GCB | $2.7 \pm 1.77$ | $63.2 \pm 2.57$ | $1.1 \pm 1.0$ |
| Type III | $19.2 \pm 2.1$ | $30.7 \pm 2.5$ | $2.1 \pm 0.7$ |

- K most probable configurations: We computed the $K = 20$ most probable configurations in the Abduction Phase of Section 6.3.2.
- Unclassified samples: We classified a sample with class $c_i$ when the classifier returned a probability $P(C = c_i|\mathbf{g}) > 0.8$. Otherwise, it shall be left unclassified.

### Experimental results

The performance of this approach is analyzed with only the Anova phase and, subsequently, by application of the two phases: the Anova and the Wrapper method with abduction.

Tables 6.1 and 6.2 show the classification data with the mean number of cases assigned to each group (and the resulting 95% confidence interval) using only the Anova phase (Table 6.1) and with the two phases (Table 6.2). The tables show the classification results for the training and test datasets using a *leave-one-out cross-validation* procedure [108] to estimate accuracy.

### Results comparison

There are several proposed classifiers ([194], [124], [8]) for the dataset given in [5]. This dataset contains 42 samples (21 GBC and 21 ABC). They attempt to differentiate between two kinds of DLBCL: *Germinal B-cell like* (GBC) versus *Activated B-cell like* (ABC). However, there are not too many proposed classifiers

Table 6.2: Mean number of cases classified in each group, using the two phases.

| **Training Dataset** | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclassified |
| ABC | $38.9 \pm 2.0$ | $0.6 \pm 0.5$ | $2.5 \pm 1.03$ |
| GCB | $0.7 \pm 0.59$ | $63.3 \pm 1.86$ | $3.0 \pm 1.26$ |
| **Test Dataset** | | | |
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclassified |
| ABC | $32.7 \pm 2.73$ | $3.5 \pm 1.27$ | $4.8 \pm 1.46$ |
| GCB | $3.2 \pm 1.64$ | $58.8 \pm 3.0$ | $5.0 \pm 1.4$ |
| Type III | $15.3 \pm 2.2$ | $29.1 \pm 2.1$ | $7.6 \pm 2.62$ |

with the new dataset introduced by Rosenwald et al. [152] (274 cases). One of the best classification results can be found in [189]. This paper shows a statistical model based in a *lineal predictor score* (LPS) which is applied to the clustering proposed by Rosenwald et al. [152]. The resulting classifier contains 27 genes. If there is no class with a probability higher than 0.9, then the case is left *unclassified*. The model was validated with the division of the dataset into two groups: training dataset and test dataset. Wright et al. [189] give the classification results shown in Table 6.3. The validation of the proposed classifier follows a similar procedure (see section 6.3.3).

The Anova Phase approach (Table 6.1) provides very good results. It leaves very little cases unclassified and the error predictions are low (around 5 cases). However, the 80 genes selected in this phase appear to be excessive if relevant biological information is sought.

The second approach proposed, the Wrapper method with an Abduction phase, starts with the set of selected genes of the Anova phase. If the results of this classifier (Table 6.2) are compared with those of Wright et al. [189] (Table 6.3), it can be said that the worst results are obtained. However, we obtain a very low number of selected genes (around 7 versus 27). In addition, our model was validated by means of 10 different partitions of the dataset. This endows our classifier with better reliability. It should be pointed out that in some of the 10 experiments we obtained better performance in both approaches than that obtained by Wright et al. [189].

Table 6.3: Number of cases classified in each group with Wright's classifier [189]

| **Training Dataset** | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclassified |
| ABC | 37 | 1 | 4 |
| GCB | 1 | 58 | 8 |

| **Test Dataset** | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclassified |
| ABC | 38 | 1 | 2 |
| GCB | 2 | 57 | 8 |
| Type III | 14 | 18 | 25 |

In a more recent paper Lossos et al. [125] proposed a model with 6 genes (LM02, BCL6, FN1, CCND2, SCYA3 and BCL2) to predict the survival rate in DLBCL. The selection is based on analysis of 36 genes whose expression had been reported to predict survival in DLBCL in previous studies based upon biological knowledge. The target of our model is different, but we believe that the genes we obtain must be similar, because each class has a distinct survival rate. We found that the Anova phase selected genes LM02 and BCL6 in 60% of the runs, and gene CCND2 in 20%. When the genes were selected in the Anova phase, they appeared in the final set of features in 33% (LM02 and BCL6) and 100% (CCND2) of the runs. Thus, it can be concluded that we selected biologically relevant genes, although biological information was not used in our procedure.

## 6.4 Some Improvements in Preprocessing and Variable Elimination

In this section, we present two significant improvements for gene selection with wrapper methods: the first one consists of a fixed ranking of genes; and the second involves the application of a method for elimination of irrelevant genes, in which the irrelevance criteria is conditioned to the selected feature set of the wrapper method. These approaches are validated with the *Diffuse Large B-Cell Lymphoma* subtype classification problem (Section 5.3). These two changes constitute an important improvement in the computational cost and the classification accuracy of wrapper methods for this domain.

The remainder of the section is organized as follows. Section 6.4.1 analyzes the importance of fixing a hierarchical ranking of genes for the performance of the model and for reducing the search space. Section 6.4.2 shows the algorithm for removing irrelevant genes based on a new heuristic. And finally, Section 6.4.3 shows the experiment results, comparing these with the results of Wright et al. [189] and with the previous approach presented in Section 6.3.

### 6.4.1 Gene ranking in wrapper search

In this section we show how the use of a given ranking of genes can be used to improve the accuracy of the classification and to reduce the search space for wrapper methods.

#### Description of the proposed gene rankings

In the wrapper algorithm described in Section 6.2, it is possible to find in an $l$ step that there are several genes $G_{\max}$, so that $G_{\max} = arg_G \max\{Acc(\mathbf{F}_l \cup \{G\})\}$, producing the same increment in classification accuracy, and therefore they are all candidates for inclusion in the set $\mathbf{F}_{l+1}$. In the domain of DNA microarrays (datasets with a high number of genes and few samples), this situation is very common, due to the big difference in the proportion between variables and samples. In particular, in the last steps of the wrapper search, the number of

candidate genes producing the same increment in classification accuracy is very high.

In order to provide a criterion to select one of these candidate genes, we propose a previously set ranking thereof. When there are several candidate genes $G_{\max}$, the one with the highest ranking is selected. This will greatly influence the accuracy of the classification, as will be demonstrated in the experimental work. Three methods are used to establish the most suitable one:

**Random Ranking:** The feature is randomly selected from among the ones producing the same accuracy.

**Anova Ranking:** The set of genes is ordered according to a filter measure, from higher to lower values. The measure considered is the Anova coefficient, which is calculated with a standard one-way analysis of variance with respect to the class variable (Section 6.3.1). The genes with a high Anova coefficient present a statistical significant difference between the means of their values for each class.

**Accuracy Ranking:** If a given classifier is trained using only one gene and a leave-one-out cross validation scheme over the training data set is employed, the accuracy of the classifier in relation to a concrete gene in the training data set is computed. With this score, the whole set of genes can be sorted, from higher to lower accuracy levels.

Section 6.4.3 shows that the classification accuracy of Wrapper methods varies meaningfully depending on the ranking method used. In particular, accuracy ranking produces the best results.

### Reducing the search space in wrapper methods using gene ranking

In this section we show how the ranking of genes can be used to reduce the search space in wrapper methods, without any significant loss of accuracy in the classification.

The method is based on limiting the search of the gene $G_{\max}$ in step $l$ to the set of the first $t$ genes in a given ranking, where $t$ is a fixed integer constant.

This modification reduces the complexity of the construction of the classifier from $O(T^2 \cdot \eta^2 \cdot n)$ to $O(T^2 \cdot \eta^2 \cdot t)$, in a database with $T$ samples (cases) and $n$ genes. The value $\eta$ represents the maximum number of variables selected by the wrapper algorithm. This value is normally much lower than $n$. Furthermore, the computation cost of the ranking of the genes needs to be added. This cost is the following:

**Anova Ranking** The cost of computing this ranking is $O(T \cdot n)$.

**Accuracy Ranking** Now a cross validation must be performed to estimate the accuracy for each variable. The resulting cost is $O(T^2 \cdot n)$.

In this way, the complexity of the wrapper algorithm no longer depends on the number of genes $n$ in the dataset. The number of genes only has influence in the ranking stage. The reduction of the search space for new genes $G_{\max}$, in step $l$ of the algorithm, does not cause loss in the accuracy classification, as will be shown in the experimental evaluation of Section 6.4.3. The resulting FSS wrapper algorithm is as follows:

**Algorithm 10** *Limited Forward Sequential Selection (LFSS)*

*Make* $F_0 = \emptyset$, $l = 0$

*While* ($\mathbf{G} \neq \emptyset$ *and* $Max\{Acc(\boldsymbol{F}_l), Acc(\boldsymbol{F}_{l-1}), \ldots, Acc(\boldsymbol{F}_{l-q+1})\} \geq Acc(\boldsymbol{F}_{l-q})$)

- $\mathbf{G^t} = \{G_i \in \mathbf{G} : Order(G_i) \leq t\}$
- $\mathbf{G}_{\max} = \{G_{l_1}, \ldots, G_{l_p}\} = args_{G_i} \max\{Acc(F_l \cup \{G_i\}) : G_i \in \mathbf{G^t}\}$ *//All genes that maximizes the accuracy.*
- $G_{max} = arg_{G_{l_i}} \max\{Order(G_{l_i}) : G_{l_i} \in \mathbf{G}_{\max}\}$
- $F_{l+1} = F_l \cup \{G_{\max}\}$
- $\mathbf{G} = \mathbf{G} \setminus \mathbf{G_{max}}$
- $l = l + 1$;

*return* $arg_{F_i} \max\{Acc(F_i) : i \in \{1, \ldots, l\}\}$;

where, as can be seen, $\mathbf{G}_{\max}$ represents the set of the genes that obtains maximum accuracy with a ranking higher than $t$ in step $l$. That is, each one of the genes in $\mathbf{G}_{\max}$ verifies that $G_i = arg_{G_i} \max\{Acc(F_l \cup G_i) : G_i \in \mathbf{G^t}\}$, where $\mathbf{G}^t$ is the set of the first $t$ genes in the ranking given by the function $Order(G_i)$.

### 6.4.2 Elimination of irrelevant genes

The basic technique for removing irrelevant genes with a wrapper method is known as *Backward Sequential Elimination* (Section 2.2.2). This method begins with the complete set of features and successively removes the ones found to be irrelevant. In [4] no evidence is found to indicate that this method is better than *Forward Sequential Selection*. Subsequent research [109; 120] develops new variants of the method. These obtain better accuracy rates, but the complexity of the algorithms is still prohibitive when there are too many irrelevant variables.

#### Irrelevant features

There are several possible definitions for relevant and irrelevant variables (see for example [6; 98]). All these definitions are based upon the correlation factor among the states of the variable to be considered and the different values of the class variable. In this section, we propose a new heuristic method for defining irrelevant variables.

Let us denote by $M$ a classifier model over a set of predictive features $\mathbf{Y} \subset \mathbf{X}$, built with a dataset $D$ with $T$ instances. Assuming that $C_Y^M = (s_1, s_2, \ldots, s_T)$ is a *classification vector* that determines whether classifier $M$ classifies well each of the cases in dataset $D$ using only the features of $\mathbf{Y}$. In a classification vector $C_Y^M$, $s_i = 1$, if the class of case $i$ is correctly found, and $s_i = 0$ otherwise. Let us now define a relation order between two classification vectors:

**Definition 1** *If $r \in [0, 1]$ is a given input parameter and $C_{\mathbf{Y}}^M = (s_1, s_2, \ldots, s_T)$ and $C_{\mathbf{Y}'}^M = (s'_1, s'_2, \ldots, s'_T)$ are two classification vectors obtained using two sets of features $\mathbf{Y}$ and $\mathbf{Y}'$ respectively, then:*

$$C_{\mathbf{Y}}^M \leq_r C_{\mathbf{Y}'}^M \quad if \quad \frac{P}{T} < r$$

*where $P$ is the number of samples that are correctly classified by the classifier $C_{\mathbf{Y}}^M$ and not correctly classified by the classifier $C_{\mathbf{Y}'}^M$. Obviously, $0 \leq P \leq T$.*

The previous definition indicates that $C_{\mathbf{Y}}^M \leq_r C_{\mathbf{Y}'}^M$ if the number of samples correctly classified into $C_{\mathbf{Y}}^M$ and not in $C_{\mathbf{Y}'}^M$ are below a given rate $r$. Now we can define an irrelevant feature in the following way:

**Definition 2** *Feature $X_i$ is irrelevant with respect to a set of features $\mathbf{Y}$ if $C^M_{\{X_i\}} \leq_r C^M_{\mathbf{Y}}$.*

Thus, a feature $X_i$ is irrelevant with respect to set $\mathbf{Y}$ if the cases correctly classified using a classifier with only the feature $X_i$ are included in the set of cases correctly classified with a classifier with the set of features $\mathbf{Y}$. The cases correctly classified could reach the $r\%$ of the total. The basic intuition idea is to seek new features classifying the cases that were incorrectly classified by current features $\mathbf{Y}$. The inclusion is not strict and there is a rate $r$ of allowed exceptions.

### A wrapper gene selection approach based on elimination of irrelevant genes

Herein we propose a new approach to wrapper search for gene selection. At each step $l$ of the wrapper algorithm, the irrelevant genes with respect to the genes included in the classifier are now eliminated. This procedure is conducted prior to the search for a new gene $G_{\max}$. Thus, irrelevant genes are not removed a priori as in [4; 109; 120], and this is based on the search process of the wrapper algorithm. This process reduces the complexity of the wrapper algorithm, and obtains better accuracy rates, as we will show in Section 6.4.3.

The wrapper algorithm that includes this new improvement and the ones specified in Section 6.4.1 is the following:

**Algorithm 11** *Limited Forward Sequential Selection with Variable Elimination (LFSS-VE)*

*Make* $F_0 = \emptyset$, $l = 0$

*While (*$\mathbf{G} \neq \emptyset$ *and* $Max\{Acc(\boldsymbol{F}_l), Acc(\boldsymbol{F}_{l-1}), \ldots, Acc(\boldsymbol{F}_{l-q+1})\} \geq Acc(\boldsymbol{F}_{l-q})$ *)*

- $\mathbf{G^t} = \{G_i \in \mathbf{G} : Order(G_i) \leq t\}$
- $\mathbf{G}_{\max} = \{G_{l_1}, \ldots, G_{l_p}\} = args_{G_i} \max\{Acc(F_l \cup \{G_i\}) : G_i \in \mathbf{G^t}\}$ *//All genes that maximizes the accuracy.*
- $\mathbf{G}_{\max} = \{G_{l_1}, \ldots, G_{l_p}\}$
- $G_{max} = arg_{G_{l_i}} \max\{Order(G_{l_i}) : G_{l_i} \in \mathbf{G}_{\max}\}$
- $F_{l+1} = F_l \cup \{G_{\max}\}$;

- *Remove* $G_{\max}$ *from the global set of features* $\mathbf{G}$
- *Remove* $G_i \in \mathbf{G}$ *if* $C^M_{\{G_i\}} \leq_r C^M_{F_{l+1}}$ *($G_i$ is irrelevant with respect to $F_{l+1}$)*
- $l = l + 1$

*return* $arg_{F_i} \max\{Acc(F_i) : i \in \{1, \ldots, l\}\}$

In the previous algorithm, the meaning of $\mathbf{G}_{\max}$ and $Order(G_i)$ is the same as in Algorithm 10. The loop now contains an additional stopping condition: if set $\mathbf{G}$ is empty. The computational cost of this algorithm is low: $O(T \cdot n)$ where $T$ is the number of samples in the dataset and $n$ the number of genes.

### 6.4.3 Experimental evaluation

#### Experimental setup

We validated the proposed approaches with two different datasets for the *Diffuse Large B-Cell Lymphoma* subtype classification problem [5]:

**D-Alizadeh:** This data set was taken from [5]. It contains 348 genes with 42 samples. There are two classes: GCB and ABC with 21 samples each one.

**D-Wright:** This dataset was taken from [189]. This dataset contains 8503 features (clones). Class GCB contains 134 samples and class ABC contains 83.

The validation of the classifier for **D-Alizadeh** is performed with the leave-one-out (LOO) cross validation method [168], due to the low number of samples of this dataset. For **D-Wright**, the dataset was randomly partitioned into two parts of equal size: the training and test datasets. The number of features in **D-Wright** is reduced by means of a previous filter (Section 6.3.1) based on one-way analysis of variance for each feature. This filter method is employed in order to make possible the evaluation of traditional wrapper methods in this big dataset, because this evaluation is impossible with its 8503 features. This whole process is repeated ten times, and 10 training datasets and 10 testing datasets are therefore obtained and the mean of the ten evaluations is the final evaluation result. This

Table 6.4: Baseline Results using the whole set of genes.

| **D-Alizadeh** | |
|---|---|
| N of Genes | $348 \pm 0.0$ |
| LOO Accuracy Rate | $97.6 \pm 0.7$ % |
| LOO log-likelihood | $-0.61 \pm 4.9$ |

| **D-Wright** | |
|---|---|
| N of Genes | $78.7 \pm 4.4$ |
| Test accuracy rate | $94.1 \pm 1.3$ % |
| Test log-likelihood | $-0.53 \pm 0.15$ |

specific evaluation scheme was used in order to compare with the results of [189] and of the previous approach of Section 6.3.

The parameters established in the implementation of the approaches in Sections 6.4.1 and 6.4.2 are the following:

- Stop Condition of FSS Algorithm. Parameter $q = 2$ (Section 6.2). That is, the FSS algorithm will stop if there are two iterations without an improvement in classifier accuracy.
- Wrapper Search Limit. Parameter $t = 10$ (Section 6.4.1). That is, the FSS algorithm only searches in the first ten ranked variables.
- Irrelevant Condition. Parameter $r = 0.02$. (Section 6.4.2). That is, a feature is irrelevant if the percentage of cases that are correctly classified using its information using its information, and that were incorrectly classified with the current set of variables, is lower than 2%.
- Accuracy Ranking. (Section 6.4.1). This is the chosen ranking in all the cases, except when another ranking is specified.

### Baseline experimental results

We obtained the results shown in Table 6.4 using a Gaussian Naive Bayes classifier including all the present genes for **D-Alizadeh** and all the genes in the reduced **D-Wright**.

### Experimental results: wrapper dependence of the feature ranking

With the use of the wrapper search algorithm described in Section 6.2, we then performed three distinct runs of this algorithm using the three ranking methods of Section 6.4.1. The results are shown in Table 6.5.

Table 6.5: Evaluation of **Algorithm LFSS** with three different gene rankings.

| Data Base | | Random Ranking | Anova Ranking | Accuracy Ranking |
|---|---|---|---|---|
| **D-Alizadeh** | Accuracy | $80.9 \pm 4.9$ | $81.0 \pm 4.9$ | $92.8 \pm 2.1$ |
| | log-like. | $-0.39 \pm 0.2$ | $-0.74 \pm 1.42$ | $-0.31 \pm 0.30$ |
| | N Genes | $4.3 \pm 0.5$ | $3.2 \pm 0.1$ | $3.8 \pm 0.5$ |
| | N Eval. | 74.900 | 77.790 | 82.300 |
| **D-Wright** | Accuracy | $88.9 \pm 0.6$ | $91.0 \pm 0.4$ | $89.1 \pm 0.5$ |
| | log-like. | $-0.41 \pm 0.15$ | $-0.35 \pm 0.1$ | $-0.40 \pm 0.13$ |
| | N Genes | $8.0 \pm 3.2$ | $9.0 \pm 5.1$ | $7.6 \pm 4.0$ |
| | N Eval. | 8.002 | 8.630 | 7.709 |

Table 6.6: Evaluation of **Algorithm LFSS** and **Algorithm LFSS-VE**

| Data Base | | Algorithm LFSS |
|---|---|---|
| **D-Alizadeh** | Accuracy | $92.8 \pm 2.1$ |
| | log-like. | $-0.36 \pm 0.44$ |
| | N Genes | $3.8 \pm 0.3$ |
| | N Eval. | 2840 |
| **D-Wright** | Accuracy | $91.8 \pm 0.4$ |
| | log-like. | $-0.28 \pm 0.07$ |
| | N Genes | $7.8 \pm 3.0$ |
| | N Eval | 1080 |

**(a)**

| Data Base | | Algorithm LFSS-VE |
|---|---|---|
| **D-Alizadeh** | Accuracy | $95.2 \pm 1.4$ |
| | log-like. | $-0.08 \pm 0.03$ |
| | N Genes | $5.4 \pm 0.1$ |
| | N Eval. | 1882 |
| **D-Wright** | Accuracy | $93.0 \pm 0.4$ |
| | log-like. | $0.25 \pm 0.07$ |
| | N Genes | $8.1 \pm 5.6$ |
| | N Eval | 1018 |

**(b)**

Comparing with Table 6.4, we can see that the accuracy rate increases and that the *log-likelihood* decreases with the ranking introduction in both datasets.

**Experimental results: introduction of a ranking limit in the feature space of the wrapper search.**

Table 6.6 (a) shows the results for Algorithm LFSS choosing new features only among the $t$ first ones in the given ranking (Section 6.4.1). Comparing with Table 6.5, we can see that there is a significant improvement in accuracy and *log-likelihood* with respect to classic wrapper search (**Random Ranking** column in Table 6.5) in both datasets. Secondly, there is a significant reduction of the number of evaluations between the two algorithms, 96% in **D-Alizadeh** and 87% in **D-Wright**. In addition, one can see that these improvements are not influenced by the number of selected genes, because they are similar in all three cases.

Table 6.7: Evaluation of Algorithm LFSS-VE with Anova and Accuracy Rankings

| **Data Base** | | **Algorithm LFSS-VE** with Anova Ranking | **Algorithm LFSS-VE** with Accuracy Ranking |
|---|---|---|---|
| **D-Alizadeh** | Accuracy | $88.1 \pm 3.3$ | $95.2 \pm 1.4$ |
| | log-like. | $-0.59 \pm 1.43$ | $-0.08 \pm 0.03$ |
| | N Genes | $3.9 \pm 0.1$ | $5.4 \pm 0.1$ |
| | N Eval. | 2.461 | 1.882 |
| **D-Wright** | Accuracy | $90.7 \pm 0.5$ | $93.0 \pm 0.4$ |
| | log-like. | $-0.31 \pm 0.08$ | $-0.25 \pm 0.07$ |
| | N Genes | $7.6 \pm 2.7$ | $8.1 \pm 5.6$ |
| | N Eval. | 885 | 1.018 |

### Experimental results: gene elimination

Table 6.6 (b) shows the results of applying Algorithm 11 of Section 6.4.2 (elimination of irrelevant variables). Comparing with Tables 6.5 and Table 6.6 (a), it can be seen how Algorithm LFSS-VE improves the accuracy rate and the *log-likelihood* of both datasets. We also obtained a reduction of the number of evaluations.

### Experimental results: accuracy order vs Anova order

The results of Table 6.7 show that Algorithm 11 performs much better with the accuracy ranking than with the ranking based on Anova. The introduction of the variable elimination mechanism, however, is positive for both rankings.

### Results comparison

There are several classifiers proposed in the literature [8; 124; 194] for the dataset **D-Alizadeh**. However there are not too many classifiers proposed for the dataset **D-Wright** introduced by [152]. Perhaps the best classification results can be found in [189]. In [189] a statistical model is shown which is based on a *lineal predictor score* (LPS) applied to the clustering proposed by [152]. The resulting classifier contains 27 genes. If there is no class with a probability higher than 0.9, then the case is left *unclassified*.

As can be seen in Table 6.8 and Table 6.9, the results of Algorithm LFSS-VE are better than those of the previous approach of Section 6.3. This classifier selects a similar number of genes (8.1 versus 7.0).

Table 6.8: (a) Classifier of [189] (b) Approach of Section 6.3

| Training Dataset | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclass. |
| ABC | 37 | 1 | 4 |
| GCB | 1 | 58 | 8 |

| Test Dataset | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclass. |
| ABC | 38 | 1 | 2 |
| GCB | 2 | 57 | 8 |

**(a)**

| Training Dataset | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclass. |
| ABC | 38.9 | 0.6 | 2.5 |
| GCB | 0.7 | 63.3 | 3.0 |

| Test Dataset | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclass. |
| ABC | 32.7 | 3.5 | 4.8 |
| GCB | 3.2 | 58.8 | 5.0 |

**(b)**

Table 6.9: Classifier of Algorithm LFSS-VE with Accuracy Rankingwith cutoff for unclassified equal to 0.9 .

| Training Dataset | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclass. |
| ABC | 37.3 | 1.0 | 3.7 |
| GCB | 0.5 | 60.0 | 6.5 |

| Test Dataset | | | |
|---|---|---|---|
| **True class** | **Predicted class** | | |
| | ABC | GCB | Unclass. |
| ABC | 32.7 | 1.3 | 7.0 |
| GCB | 1.7 | 57.4 | 7.9 |

On the other hand, the results of Algorithm LFSS-VE are similar to those of [189], but the latter obtains a lower number of genes, 8 versus 27, and our validation is performed in ten distinct partitions of the dataset in relation to the only evaluation of the classifier of [189]. Indeed, there are better results than [189] in several of the ten evaluations of our classifier.

## 6.5 Conclusions and Future Work

When treating with gene expression data (a very high number of features and a low number of instances), the difficulty does not involve finding a complex classification model, but rather reducing the high number of features.

In this chapter, we make two different proposals for dealing with the peculiarities of this kind of data.

In Section 6.3, we proposed a filter-wrapper approach with an abduction phase. A filter approach was first detailed in Section 6.3.1. We should highlight the great capacity of the Anova function and of the correlation coefficient to remove redundant and irrelevant genes. In the experiments of Section 6.3.3, the initial 8503 genes were reduced to 75, obtaining accuracy rates of 94.1% while biologically relevant genes were maintained. Another problem of DNA microarrays is the low number of available samples, which can give rise to overfitting of classifiers. In order to deal with this, we employed a wrapper methodology combined with the use of an abduction method (most probable explanation) to predict a robust set of genes, Section 6.3.2. The experimental work shows that this initial approach provides a high accuracy level. The results of this wrapper model with an abduction phase are similar to the results of [189] for the DLBCL classification problem.

In Section 6.4, we made some additional changes to the wrapper search in order to improve performance and reduce its computational complexity, avoiding the need to use fast filter methods. As can be seen in the experimental results (Section 6.4.3), the gene ranking and the wrapper search in only the first $t$ genes constitute an excellent method for reducing the computational cost of the wrapper search without any loss in the classification accuracy rate. Furthermore, the introduction of a new heuristic for irrelevant gene elimination depending on the

wrapper search process presented very good behaviour when applied to the *Diffuse Large B-Cell Lymphoma* classification.

A future line of work involves the validation of theses models with other datasets, for example ones dealing with breast cancer, colon cancer, leukemia, etc. In addition, the use of other classification models with more complex structures is another important issue that we wish to explore in the future.

# Part IV

# Applications to Information Retrieval

# Chapter 7

# Information Retrieval in Context

## 7.1 Introduction

The term *information retrieval* has many different meanings. Just picking up a *post-it* from your work desk to read what you wrote is a form of information retrieval. From a formal point of view, *information retrieval* might be defined thus [38]:

**Information retrieval (IR)** is finding documents of an unstructured nature, usually text, that meet an information need from within large collections, usually stored in computers.

Thus, *information retrieval* used to involve the activity of a few specialists, such as reference librarians. In the last decade, the *World Wide Web* has completely changed the way people pursue information. The emergence of popular web search engines such as *Google, Yahoo, Microsoft Live, etc.* has provided the most sophisticated *information retrieval techniques* to satisfy the information needs of hundreds of millions of people every day.

IR can also embrace other kinds of information problems different from what is indicated above. The term *unstructured data* defines a kind of data that is not directly related to the underlying representations used by a computer. One example of *structured data* would be the relational database widely used by companies to store very different kinds of data. But, in reality, practically no data are

completely unstructured, particularly for all written text data containing a latent linguistic structure. Most text data have a structure, such as titles, headings and paragraphs, which is usually defined in web documents by explicit markups. *Semi-structured search* can be seen as an information retrieval approach that exploits this partial structure of text data: for example, search documents where the title is devoted to supervised classification.

The IR field also involves supporting user browsing, filtering some kind of documents or further processing certain groups of retrieved documents. Two main tasks, closely related to the Machine Learning field, are document clustering and document classification. Document clustering consists of grouping documents according to their content or to other features, as a latent approach to structuring a large set of documents. Moreover, document classification deals with the problem of assigning a predefined label to each document. It is widely used in many real IR applications, one very common example being automatic email spam classification.

Another distinction of information retrieval systems involves the scale at which they are built. Three prominent scales are commonly distinguished. In web search, the IR systems has to deal with billions of documents, stored in millions of computers and requested by millions of user searches. Consequently, special and specific issues have to be tackled in order to gather text data for indexing, to work efficiently with this huge quantity of data and to handle specific aspects of web documents, such as exploitation of html markups. These systems should also be able to detect illegal page content manipulations in order to boost the ranking in web search engines.

The other end of this scale involves personal information retrieval. In recent years, operating systems have plugged IR systems into their core software: Apple's Mac OS X Spotlight or Windows Vista's Instant Search. Email applications not only provide email search but also email classification, such as spam mail classification or automatic means for assigning predefined labels. The specific issues handled by these IR systems include treatment of the different document types in a common personal computer and rendering the system easy to maintain and sufficiently lightweight in terms of computation and storage resources, in order to avoid annoyance to users.

Between these two previous scales lies the enterprise and domain-specific search, where document collection involves the internal documents of a corporation, patent databases or research articles on computer science. In these cases, all documents are stored in a centralized file system and dedicated machines are employed to run the IR system.

In short, IR encompasses a wide area of research. It involves several very different issues related to supplying information for people's day-to-day information needs.

## 7.2 The Notion of Context

*Context*, once a promising concept, appears to have become a cause of confusion in Information Retrieval (IR) and related fields. This confusion would seem to be due to the scope of concept [40]. In the narrowest sense, the scope of context in search can refer to elements surrounding words [e.g., 9; 61]. For example, by looking at the words co-occurring with query terms in documents, one might understand a searcher's underlying information need that was originally expressed by few words. A wider scope of context is the history of interaction [e.g., 62]. If we look at iterative searches as a conversation between a searcher and an IR system, then it makes sense to consider the past dialogue in order to understand the current discourse. Similarly, by exploiting past search activities and interaction with a search interface, one might better understand a searcher's underlying information need.

But, as was pointed out by [92], *IR research is now conducted in multi-media, multi-lingual, and multi-modal environments, but largely in a context-free manner*. However, it is well known that the process of retrieved information strongly depends on time, place, interaction, task, and a wide range of factors that are implicit in the user interaction and the environment: *the context*. All this contextual information can be exploited to restrict the information space and, thereby, to boost the performance of IR systems.

Recently, work tasks have been studied as a promising context [e.g., 104]. A work task can be any information activity people perform on a day-to-day basis motivating search activities [20]. Thus, this can be seen as an increase in the

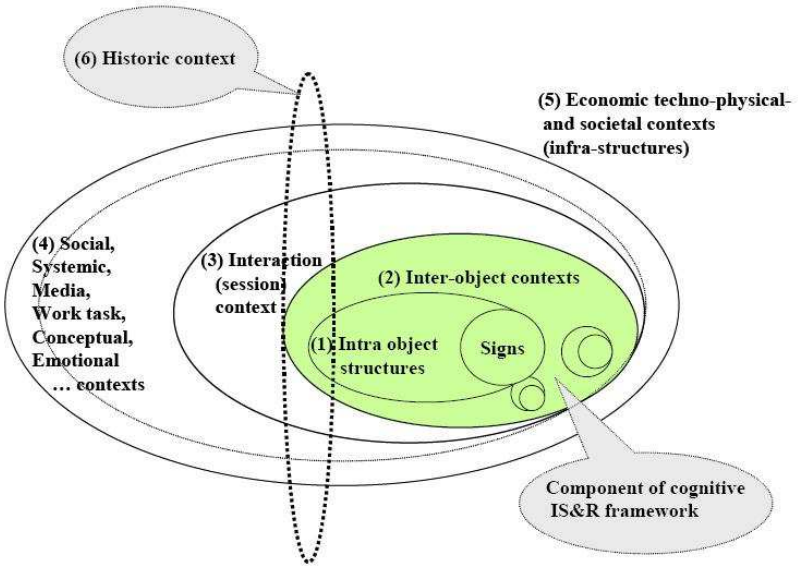


Figure 7.1: Nested model of context stratification for IR [92].

scope of the concept. It has been suggested that work tasks can indicate the relevance of the genres of documents [63]; thus, the document genres can also be seen as a context. In the field of Human Computer Interaction (HCI), environmental factors such as location and time of activities have been investigated in the development of context-aware systems [52]. Similarly, the devices that people use in searches are seen as a contextual factor to be considered, for example, for the presentation of search results on mobile devices [102].

Figure 7.1, taken from [92], shows a stratification of contexts related to IR engines and systems. This stratification involves the traditional content features of information objects (i.e. words placed inside paragraphs), hyperlinks and interaction features (i.e. mouse movements and clicks during search session or daily-life task situation). The main hypothesis states that by considering all the information associated with these contexts, the next generation of IR systems will outperform traditional context-free search engines.

Whether or not there is consensus regarding the scopes of context, researchers appear to agree with the importance of context in IR. One implication of the widening scope of context is that there can be many relevant contextual factors when we study information seeking in particular environments or when we develop

a context-aware search model/application.

These so-called context-aware systems [28] constitute a new area of IR systems that attempts to overcome the limitations of traditional personalization systems. In this scope, the term *context* can take many different meanings and no widely accepted definition exists [56].

Several context-aware retrieval systems exist in the literature and most of them are based on statistical models for combining preceding queries and click-through documents with the current query in order to boost the ranking of documents [14; 61; 123]. But optimal IR systems should exploit as much contextual information as possible whenever available. *Relevance Feedback* provides a common solution to this problem. However, its effectiveness is limited in real systems, basically because of the reluctance of users to provide such information.

For this reason, *implicit feedback* has recently attracted much attention [32; 106; 180]. Needs for complex information involve the submission of different queries by the user and viewing of different ranked documents before the user's information need is met. In such an interactive scenario, useful information naturally emerges and is available to the IR system beyond the initial user query and the document collection. Generally, an interaction history track can be exploited by the retrieval system, including previous queries, click-through documents and how users read these documents.

## 7.3 Context in Search

The integration of *context* ideas into IR systems is not new, although the notion of context can be very different across the different approaches.

Lawrence et al. [123] offers a thorough review of the employment of context in Web search. Explicit information can be submitted to a search engine as a category restriction. This category can help to disambiguate a query and improve the results. For instance, given the query "java", possible categories are "island" or "programming language" [75].

To the contrary, other systems automatically infer context information by analyzing the documents displayed to the user [29]. These tools face difficulties when documents are too long and involve several topics.

Another family of IR systems considers the notion of context as a group of information requests generated by the user. Thus, context become a form of personalization and these systems keep track of a user's previous queries and click-through documents [14].

Other approaches for integrating context into searches involve the use of domain-specific search engines [123]. IR systems that search the Invisible Web (i.e., whose information content is not indexed by traditional search engines) could become very useful, as they contain huge amounts of information within a very defined domain.

As can be seen, there are many interpretations of the notion of context and how this notion is implemented in IR systems. This variety probably arises from the difficulties involved in this problem and determines the need for further research in order to provide better knowledge of what context is and how it can be exploited.

## 7.4 Interactive Information Retrieval

In the last few decades, most efforts in IR research have focused on methods for matching text representations with query representations. However, IR researchers have recently addressed the task of understanding the role of the user in IR. The basic hypothesis is that knowledge of how users interact with IR systems constitutes an effective way to improve the performance of theses systems. Therefore, the line of research involving users in the process of consulting an IR system is known as interactive information retrieval (IIR). IIR can be seen as a limited, initial and direct way to implement context-aware IR systems, because they consider certain aspects of the broad spectrum of factors detailed in Section 7.2.

Traditional models of IR hardly consider the dynamic nature of the interaction phenomenon between users and IR systems. Herein we detail four basic models exploring the underlying dynamics of interactive IR:

- Saracevic's (1997) stratified model of interactive IR [157]: This model considers multiple dimensions of user participation in IR processes. That is,

this model accounts for user knowledge, along with its environment and situation. It is partially capable of describing the complexities surrounding the interaction between users and IR systems.

- Belkin's (1996) episodic model of IR interaction [11]: This model is based on user commitments to their anomalous states of knowledge. It basically considers that users carry out search tasks with different degrees of knowledge in relation to the subject of their searches.

- Spink's (1997) interactive feedback and search process model [166]: This model is based on the description of the interactive feedback and search processes regarding the complexities and cyclical nature of IR interaction. It includes time as a critical factor in IIR.

- Ingwersen's (1996) cognitive model of IR interaction [89]:

  In this model, Ingwersen attempted to synthesizes many of the aspects considered by the above models. He attempted to model interactive IR from a global perspective, stating that a wide range of factors, such as social environment influences the IR process.

Each of these models attempts to provide an alternative to traditional models of information retrieval. The limitations of these traditional approaches have been shown by IIR research. Mainly because these approaches do not consider the complexities of interaction among humans or among users and IR systems or with respect to explicit or implicit feedback provided by users [165; 166]. Aiming to face these limitations, a growing number of researches started to address problems associated with the interaction dynamics in IR, most of which are summarized in the four models enumerated above.

# Chapter 8

# Investigating the impact and dependency of contextual factors in relevance modelling

This chapter presents an approach for measuring the effectiveness of contextual factors to predict the relevance of click-through documents. The approach enables us to investigate the impact of a range of aspects such as topics, search interface features, task complexity, search stage, and search experience on relevance modelling.

## 8.1 Motivation

A prominent role of context in information retrieval (IR) involves improving modelling of document relevancy. Estimating the potential impact of contextual factors can facilitate the development of new search models which exploit context. However, eliciting promising contextual factors from a number of potentially relevant factors poses a challenging task, since there is no easy way to measure the potential effect of context. Dependency of contextual factors is even more difficult to measure. This is important, however, as search activities are shaped not by one single contextual factor, but rather by multiple ones. [91].

In order to address these generic research problems on IR in context, this chapter proposes an approach for measuring the potential impact of contextual

factors using aggregated relevance judgements that are made in controlled environments. The contributions of this chapter are as follows. First, we propose an approach which enables us to measure the impact of context and to find effective features for relevance modelling. Second, we demonstrate that the strength of context in people's relevance judgements can be captured by query-independent document features. Third, we provide empirical evidence that shows that more contexts can improve relevance modelling performance. Finally, we suggest a set of robust features that can be used for future work.

The remainder of the chapter is structured as follows. Section 8.2 reviews existing studies to elicit significant context. Section 8.3 discusses the methodology used in our approach to measure the strength of context. Section 8.4 provides descriptions of the experiments performed in our study. Section 8.5 presents the results of our experiments, followed by the discussion of the main findings and implications in Section 8.6. Section 8.7 concludes with the implications for future work.

## Definition of terms

This chapter uses several terms as follows.

**Context** is an element encompassing an information searching process. A context can have several instances, each of these instances will be called as **context group**. For example, *Search experience* context has two groups, more experienced and less experienced.

**Contextual relevance** refers to relevance of documents perceived by searchers in a particular context.

**Contextual document grouping** refers to a process of grouping click-through documents (along with subsequent relevance judgements) to represent a context and its contextual relevance.

**Features** refer to a set of variables extracted from retrieved objects. These are used to generate relevance models.

**Feature category** refers to a particular category to which a feature belongs.

## 8.2 Measuring the Impact of Context

A series of forums have been held to address aspects of context in information seeking and retrieval [40; 44; 45; 90; 91; 154]. The advances reported in the forums ranged from theoretical ones, such as a taxonomy of contextual features, to empirical ones, for instance, deriving new context from environments and to constructive ones, such as new applications that exploit context. Our study attempts to make a methodological advance in this area by developing a framework for measuring the impact of contextual factors. Therefore, this section discusses different approaches taken to measure the impact of context for modelling document relevancy. It should be noted that in this study we take the view on the scope of context as shown in the Ingwersen context stratification [92] (see Section 7.3 for details).

One way to examine the impact of contextual features is to investigate the factors that influenced people's relevance judgements. For example, [10] discussed two sets of semi-structured interviews carried out to establish the criteria used for judging document relevancy. The study identified ten criteria categories common to both interviews. Their results highlighted the fact that people employed non-topical factors such as quality of sources for relevance judgements. Tombros et al. [172] observed interactive search sessions to extract the factors influencing people's relevance judgements. They identified five groups of influential factors based on 24 participants performing three different search tasks on the Web. Their results suggest that non-textual elements in documents such as structure and visual features affect people's relevance assessments.

Another way to examine the impact of contextual features is to investigate their effect on searching behaviour. For example, [105] studied the effect of tasks and searchers on reading time of retrieved documents. Their experiments show a significant correlation between contextual features and searching behaviour. Reading time was found to vary across search tasks as well as individual searchers. This suggests that reading time can be difficult with regard to modelling relevance without context. A similar approach was taken by [181], who studied the effect of topic complexity, search experience, and search stage in the performance of implicit relevance feedback. Implicit feedback was used to suggest expansion

terms in the study. A mixture of measures such as subject assessments, take-up rate of suggested terms and retrieval effectiveness was used to capture the effect of the contextual features. Study thereof shows that all three factors affect the utility of implicit relevance feedback.

A different approach employed by [62] involved modelling document relevancy based on a history of interactions. They analysed a couple of dozen user interactions and explicit relevance judgements to construct predictive Bayesian models. It can be seen that the accuracy of relevance prediction of the models was used as a measure of impact in their study. One advantage of their approach is the number of variables that can be investigated. While other approaches can examine two or three factors at a time, the classifiers enable a large number of potentially effective factors to be investigated. A disadvantage is that the dependency of features in the models generated is not always clear or interpretable.

Another way to find a dependency between contextual features is to measure the frequency of their co-occurrence in search environments. For example, [63] looked at two contextual features, document genres and work tasks, to find the dependency between them. The use of documents in a software engineering workplace was analysed in their study. The experiments show that there is a significant correspondence of document genres with the types of work tasks, which suggests that one can learn relevant genres by understanding the roles and tasks of an organization.

Compared with existing studies, our work presents the following characteristics. First, as in [172], we measure the impact of context based on searchers' relevance assessments. This is because relevance judgements are a fundamental process in search, and because we are also interested in better relevance context modelling. Second, we use a probabilistic classifier to model document relevancy. This allows us to go beyond the subjective assessments or simple frequencies to measure the impact of contextual features on document relevancy. We use query-independent document features for modelling, as opposed to interaction data used in [62]. Third, our work evaluates the range of context features that have been discussed in this section ( document textual, visual/graphical, visual/typographical, layout, structural and other selective features). Finally, the approach proposed

enables us to understand the dependency of contextual features, a similar objective to [63], in the same single framework.

The next section discusses our approach for measuring the impact of context in details.

## 8.3 The Methodology: Divide and Conquer

Our approach is based on a set of relevance judgements made in controlled environments. An overview of our approach is as follows. All click-through documents recorded in a user study were seen as a dataset *full* of context. Explicit (binary) relevance judgements were obtained for all click-through documents during the study. The full dataset was then divided into subsets based on a contextual factor in question. The relevance models were generated for all sets of data, and prediction accuracy was compared to measure the impact of the context. This process is shown in Figure 8.1, where search experience was used as an instance of context.

An assumption underlying this approach was that if a context was significant, it would have an effect on people's relevance assessments that could be captured by a relevance model (a classifier in our case). The performance of relevance models for predicting document relevancy was therefore used to measure the impact of context.

The rest of the section presents our approach in greater detail. It first describes how context is represented in the methodology. The conceptual category of document features used to model contextual relevance is then shown. This is followed by a formal description of the modelling approach.

### 8.3.1 Representing context using aggregated relevance judgements

The first step for measuring the impact of context in our approach involved representing contextual relevance. While there are different ways to represent contextual relevance, we take a simple approach which grouped a set of documents that were accessed and judged by searchers in a given context. For example, for

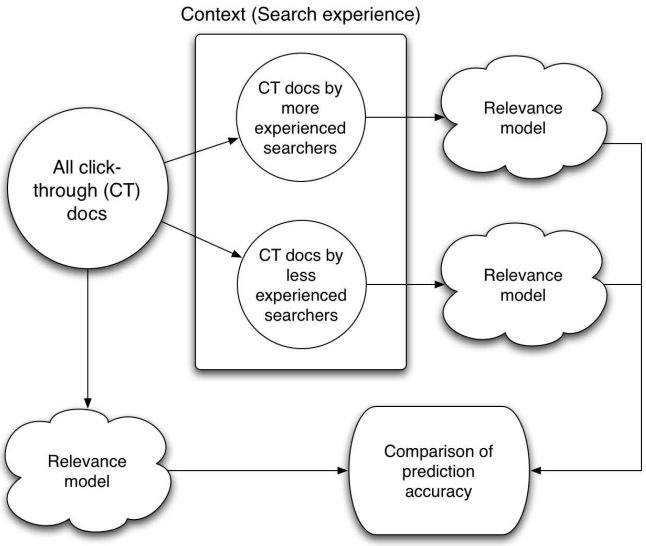


Figure 8.1: Proposed approach to measure the impact of context (e.g., search experience).

search experience context, all documents judged by searchers were divided into two groups: those that were judged by more experienced searchers, and those judged by less experienced ones (See Figure 8.1). As we will see in Section 8.5, this *contextual grouping* of documents provides us with a simple and intuitive interpretation of the impact of context. Moreover, one can apply the grouping to different representations of contextual factors such as binary data (presence or absence of a factor) or continuous data (e.g., first 5 minutes, second 5 minutes, etc.). The grouping was repeated for all contextual factors investigated in this study. Section 8.4 will show the details of the contextual factors.

It should be noted that disagreement of relevance judgements can be found in this implementation of the contextual grouping. This could cause a problem on measuring the impact of context. In this study, we chose to discard the documents from the analysis only when there was complete disagreement (i.e., 50% of judgements was relevant and the other half was non-relevant). Otherwise, relevance of documents was determined by the vote of the majority of judgements. In general, the proportion of discarded documents was very small. The details of relevance judgements are found in Section 8.4.

Table 8.1: Categories of query-independent document features.

| Feature category | Code | Size | Example |
|---|---|---|---|
| Document textual features | Text | 13 | Page length |
| Visual/Graphical features | Visual | 16 | Image area |
| Visual/Typographical features | Vis-tag | 17 | Frequency of `b` tag |
| Layout features | Layout | 14 | Frequency of `div` tag |
| Link structure features | Structure | 10 | Number of outlinks |
| Selective words | Other | 22 | "search" |
| Selective HTML tags | Oth-tag | 24 | Frequency of `script` tag |

### 8.3.2 Conceptual categories of object features

The second step of our approach involved identifying candidate features that can be extracted from retrieved documents. With the use of some informal experimentation and of literature survey, we identified over 100 document features. To increase understanding of candidate features in relevancy prediction, we then grouped them into a set of feature categories. The structure used for the categorisation is shown in Table 8.1. As can be seen, there are seven categories. An overview of the main categories is as follows.

**Document textual features:** This category consists of features that were related to textual contents of documents. The examples of features included the number of words in a document and anchor texts, number of upper-case words, number of digits, Shannon's entropy value [161] for a document and anchor texts.

**Visual/Graphical features:** This category consisted of features related to graphical and colour elements of documents. Examples of features included the number of images, dimension of images, background colour, etc.

**Visual/Typographical features:** This category was similar to the previous category in that the features were mainly visual elements of documents. However, this category included the typographical elements, such as size and colours of fonts.

**Layout features:** This category contained the features related to the layout of documents, such as number of paragraphs, tables, and alignments of elements.

**Structural features:** This category consists of features related to hyperlink and site structure of documents. The examples include the depth of document in a URL, the number of outlinks, PageRank scores.

**Other selective features:** The last two categories consisted of features that did not necessarily fit into the abovementioned categories. *Selective words in document* included the presence of selective words such as address, search, and help. Selective HTML tags included a set of HTML tags such as `form`, `object`, and `script` and their attributes.

Table 8.2 lists all the features used in our experiments. The following sections describe a methodology proposed for building a classifier, for selecting significant features, and finally, for validating the results.

### 8.3.3 Modelling contextual document relevancy

This section provides a formal account of how query-independent document features were used to model contextual relevance in our approach.

#### Methodology discussion

Firstly, we discuss the proposed methodology for measuring the strength of a context in relevance modelling.

**Language Models and query-independent features:** In this approach, contextual relevance is modelled through its probabilistic relationship with document object features, also known as query-independent features because they do not depend on the query. In order to better understand how this modelling can be derived from a general IR framework, we will make use of the *Language modelling framework* (LM) [145].

The Language modelling framework enables incorporation of query-independent features into the information retrieval task. The derivation of the LM retrieval

Table 8.2: Members of document features.

| Text (13) | Visual (16) | Vis-tag (17) | Layout (14) | Structure (10) |
|---|---|---|---|---|
| # Words | # Link Style | # b | # area | # links |
| # Different Words | % Area Web | # big | # center | # link-mail |
| # Digits | # Images | # font | # br | # In-host links |
| # Upper Case Words | % Image Area | # i | # div | # Out-host links |
| Entropy | # Background (BG) Images | # H1-6 | # map | # HTML Links |
| % Entropy | Color Foreground | # small | # hr | # Non HTML Links |
| # Words (a) | Color Background | # span | # li | Levels of URL |
| # Different Words (a) | Width of Web | # strong | # p | Page Rank (PR) Score |
| # Digits (a) | Height of Web | # u | # table | PR Score of Host Page |
| # Upper Case Words (a) | Image Disk Size | # style | # td | URL Domain |
| Entropy (a) | BG Image Disk Size | # alt | # tr | |
| % Entropy (a) | % BG Image Area | # border | # th | |
| Page length | $\mu$ Images Width | # color | # align | |
| | $\mu$ Images Height | # face | # size | |
| | $\mu$ BG Images Width | # bgcolor | | |
| | $\mu$ BG Images Height | # cellpadding | | |
| | | # title | | |
| Other (22) | | Oth-tag (24) | | |
| # email | # email (a) | # address | # action | # onmouseover |
| # address | # address (a) | # form | # method | # onmouseout |
| # tel | # tel (a) | # input | # scrolling | |
| # updated | # updated (a) | # label | # src | |
| # search | # search (a) | # object | # checked | |
| # help | # help (a) | # script | # media | |
| # sitemap | # sitemap (a) | # select | # onload | |
| # contact | # contact (a) | # meta | # onunload | |
| # contacts | # contacts (a) | # numLinkArea | # onchange | |
| # home | # home (a) | # lang | # onsubmit | |
| # languages | # languages (a) | # accesskey | # onclick | |

#: Frequency of occurrence of a feature; %: Percentage of a feature value in a page; $\mu$: Mean of multiple instances; (a): Occurrence in anchor texts.

model is estimated indirectly by invoking Bayes' rule. Thus, the probability of relevance $P(R|Q,D)$ given a query $Q$ and a document $D$ is computed as follows:

$$P(R|Q,D) = \frac{P(Q,D|R)P(R)}{P(D,Q)} \tag{8.1}$$

$$= P(Q|D,R)P(D|R)\frac{P(R)}{P(Q,D)} \tag{8.2}$$

$$= P(Q|D,R)P(R|D)\frac{P(D)}{P(Q,D)} \tag{8.3}$$

Assuming independence between queries and documents $P(Q,D) = P(Q)P(D)$ and given that $P(Q)$ is just a proportional constant:

$$P(R|Q,D) \propto P(Q|D,R)P(R|D)$$

where $P(Q|D,R)$ is the query likelihood and $P(R|D)$ is the *document prior*, that is to say, how the query-independent features of a document affect the relevance ($D$ denotes the vector of query-independent features of the document).

In Equation (8.1), we consider a strong independence assumption, but it was considered to obtain, by means of simple transformations, a final formulation with dependence on $P(R|D)$. The derivation presented in [117], which connects Language models with the probabilistic model of retrieval, took a more reasonable assumption, $Q$ and $D$ are independent under $\bar{r}$ (non-relevance), and starting from the odds-ratio of Relevance, the final relevance score is dependent on $P(r|D)/(1-P(r|D))$, which also shows the abovementioned dependence relation with the *document prior*.

In many of the common applications of LM, $P(R|D)$ is taken to be uniform and discarded from the model. However, the incorporation of prior evidence is known to boost the performance of IR systems in many different situations [43; 114]. But no accepted model as yet exists for incorporating and combining this prior evidence [15; 142].

**Justifying the probabilistic classification approach:** Some big problems arise when modelling $P(R|D)$. Firstly, the number and nature of query-independent

document features can be vary greatly and are not well defined [43; 114]. Secondly, many of the previous approaches that have attempted to model the relation of several query-independent features assume independence [114] or use a linear combination where the weights associated with each feature are set using heuristic methods. Furthermore, to our knowledge, unlike our study, no previous one handles more than one hundred features. Most of them do not employ more than 5 or 10 features. Thus, the previous approaches will probably find more problems dealing with this high number of features.

Another important issue in modelling relevance with query-independent features can be seen in the fact that not all proposed or examined features need to be valid or suitable for this purpose. Many of them will probably be too noisy or irrelevant. A recent study [141] has shown that retrieval performance can be further enhanced by selecting query-independent features depending on the query submitted to the system. Hence, the modelling relevance approach should also integrate a feature selection mechanism.

Considering these problems, the modelling approach used herein is based on the use of supervised classification models as described in Chapter 2. We argue that this approach naturally models this kind of conditional probabilistic relations and has therefore been widely employed by the Machine Learning community for this purpose.

The main disadvantage of this approach with respect to the above mentioned ones is its supervised nature, that is to say, a previous pool of data is needed to learn the model encoding the conditional distribution $P(R|D)$. This problem can be overcome because, in this study, we avail of a set of relevance judgements of a wide set of web documents compiled in controlled environments.

**Probabilistic classification:** Generally speaking, the classification problem involves the ability to predict a given feature of an object using another set of features of the same object (see Chapter 2 for details). In this concrete problem, we seek to predict the relevance (more concretely, the aggregated relevance) of a web document using its query-independent features.

Now we proceed to detail the specific notation. In the probabilistic classification paradigm, the classification problem involves two types of random variables:

**Class variable: R.** This random variable is the variable to be predicted. In this case, the class variable takes two cases $\{Relevance, Non-Relevance\}$, as the two possible predictions that can be made.

**Predictive/Attribute variables: $\mathbf{X} = \{X_1, X_2, ..., X_n\}$.** In this case, the predictive variables are the query-independent features or document features, Section 8.3.2. Thus, $\mathbf{X}$ is the set of variables described in Table 8.2.

Thus, our objective is to learn the probability distribution $P(R|\mathbf{X})$ which is estimated by applying Bayes's Theorem as follows:

$$\mathbf{P}(R|X_1, \ldots, X_n) \propto P(X_1, \ldots, X_n|R)P(R) \tag{8.4}$$

In order for the estimation of $P(X_1, \ldots, X_n|R)$ to be feasible, classification probabilistic approaches perform a factorization of this distribution using some conditional independence assumptions. The best known and simplest one is the Naive Bayes (Section 2.2.1) where all the variables are assumed to be independent if the class variable is known.

**Justifying the context division approach:** An initial natural attempt to introduce context information would involve defining a random variable $C$ taking as many cases as context groups (i.e., search-experience would be the context variable taking two cases: more-exp and less-exp) and, subsequently introducing it into the probability distribution:

$$\mathbf{P}(R|X_1, \ldots, X_n, C) \propto P(X_1, \ldots, X_n|R, C)P(R|C)$$

The problem with this approach is that the conditional independencies exploited to encode the probability distribution $P(X_1, \ldots, X_n|R, C)$ have to be maintained for the different context groups of $C$. This imposes strict restrictions with regard to modelling contextual relevance; for example, it is necessary to have the same query-independent features modelling contextual relevance and the same conditional independencies across the different context cases. However, previous works [126; 141] have shown that relevant features and their relations can vary across different queries and/or topics.

The approach we use here is related to the notion of *context-specific independencies* or *asymmetric conditional independencies* [21]. These kind of conditional independencies have been widely studied in Machine Learning literature and several models for handling them have been proposed [21; 71; 95; 170].

Firstly, we give the definitions of context-specific independencies and of conditional independencies in order to compare both definitions:

**Conditional Independence:** We say that $X$ and $Y$ are conditionally independent given $R$, if $\forall x \in X, \forall y \in Y, \forall r \in R$ the following equality is maintained:

$$P(x, y|r) = P(x|r)P(y|r)$$

**Context-specific independence [21]:** In this case, given a fixed value $c \in C$, we say that $X$ and $Y$ are *contextually independent* given $R$ and the context case $c$ if $\forall x \in X, \forall y \in Y, \forall r \in R$ the following condition is maintained:

$$P(x, y|r, c) = P(x|r, c)P(y|r, c)$$

It should be noted that if the above condition is true for all values $c \in C$, we would have the previous formulation of *c*onditional independence. In *c*ontext-specific independence the above condition only holds for some specific values of $c \in C$.

As was mentioned in Section 8.3.1, contextual relevance was represented using aggregated relevance judgements. Thus, the relevance of a document is determined by the particular context group (i.e., more-exp users) in which users judgements were made. Thus, and assuming the need to handle *context-specific independencies*, we approached the contextual relevance modelling by building different probabilistic models for each context group (i.e. one model for more-exp users and another model for less-exp users). Thus, the Equation 8.4 is newly defined as follows:

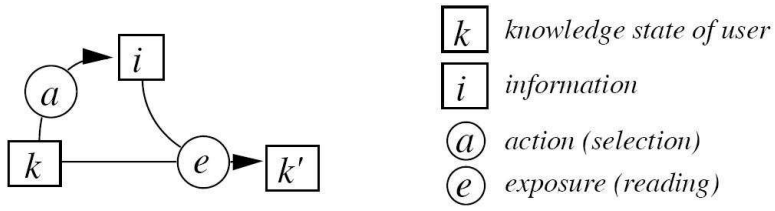


Figure 8.2: The updating of a knowledge state through the selection of, and subsequent exposure to, information. [31].

$$P_c(R_c|X_1,\ldots,X_n) \propto P_c(X_1,\ldots,X_n|R_c)P(R_c) \tag{8.5}$$

where the role of the specific context group in which the modelling is performed is now integrated into the probability distribution $P_c$ itself and the relevance is also subjected to this specific context group $R_c$ for the aforementioned reasons. As has been mentioned, to measure the impact of the context *search-experience* two models were built: $P_{more-exp}(R_{more-exp}|\mathbf{X})$ for *more-experience* users and $P_{less-exp}(R_{less-exp}|\mathbf{X})$ for *less-experience* users.

**The *a priori* probability of relevance $P(R)$:** Equation 8.5 can be interpreted from a belief updating point of view. Along these lines, this updating was expressed in *the ostensive model of developing information needs* proposed in [31]. A rough description of the knowledge updating process used in this model can be seen in Figure 8.2.

This model relates changes in the knowledge state of a user in response to information presented during information seeking activities. Extrapolating this model to our specific problem, it can be said that a user has a determined knowledge state "$k$" and takes an action "$a$" which, in our case, involves clicking on a given document link of the result list to access the web document content "$i$". The user is immediately exposed to this information through the process "$e$" and reaches another knowledge state "$k'$". Again, the user takes another action that can involve bookmarking this document as relevant, returning to the results list to click on another document of the result list or formulating another query.

In the probabilistic approach this process can be modelled considering $P(R)$ as the prior belief about the relevance of the click-through document prior to being exposed to its content. Once the user has been exposed to the content of the document (to the concrete values of the query-independent features of this document, $\mathbf{x}$ ), he/she updates his/her belief about the relevance of the document as *a posteriori* probability $P(R|\mathbf{x})$. Based on this new updated belief $P(R|\mathbf{x})$, the user then decides to bookmark or not to bookmark the click-through document as relevant. This modelling implies the Bayesian conception of probabilities as a subjective degree of belief [46] (for a review of the use of subjective probabilities in IR refers to [171]).

Once we have provided a belief updating interpretation to our modelling approach, it will become easier to understand why we opted to fix an uninformative *a priori* probability of relevance: $P(r) = P(\bar{r}) = 0.5$.

Firstly, fixing an *a priori* probability of relevance higher than 0.5 could be argued by stating that when a user clicks on a document his/her predisposition or personal belief is biased towards relevancy. However, the marginal probability of relevance that can be estimated from the data of the two user studies employed in this research does not reveal a clear trend in favour of relevant or non-relevant documents (see Table 8.4 for details). Moreover, handling datasets presenting a clear imbalance of their classes is known to deteriorate the performance of a classifier [96].

We found neither theoretical nor practical reasons to employ non uniform *a priori* probabilities of relevance, and therefore opted to learn the classification models from balanced datasets (with the same number of relevant and non-relevant documents). To address this issue, we randomly removed the samples from the larger class until the portion was balanced from the training set. Test datasets were excluded from this processing.

**Measuring the effect of context:** To measure the effect of a context we estimate how well the contextual relevance modelling given by $P_c(R_c|\mathbf{X})$ fits its real distribution. That is to say, given a concrete web document or, more precisely, the concrete values of the query-independent features for this document, the model will make a prediction about the relevance of the document and this

prediction will be compared with the real aggregated relevance judgements made by users in this context. The percentage of the correct predictions by this classification model is defined as *classification accuracy* and this measure will tell us how well contextual relevance can be modelled by document object features inside this context (the concrete method used to estimate this accuracy will be subsequently detailed in Section 8.3.3). Our assumption is the following: *the higher the classification accuracy, the stronger the context.* We will consider that a classification accuracy is indicative of contextual relevance modelling when it is statistically significant higher than 50%. We took this baseline because this is the expected classification accuracy rate if no modelling of the contextual relevance is performed. That is to say, if we assume, as discussed in the previous section, that the *a priori* probability of relevance is uniform, the expected accuracy rate will be 50%, and the modelling of contextual relevance will therefore be significant when the accuracy rate obtained by means of the knowledge of the query-independent features is higher than this threshold.

In those contexts or feature categories, when the classification accuracy is not statistically significant higher than 50%, it can be said that there is no contextual relevance modelling or, simply, no modelling.

### Methodology implementation

Once the methodology has been discussed, we will provide details of the implementation. The concrete models and approaches employed in this study were intended to be simple and standard, although some adaptations were made in order to solve some of the peculiarities relating to this problem. We point out that more specific models and approaches could be used instead of those employed here. But as an initial study, we do not explore the broader possibilities that can be applied in order to model the distribution $P_c(R_c|\mathbf{X})$.

**The classification model:** The chosen classifier model was the AODE classifier (Section 2.2.6 ). This classifier was selected for its competitive performance, as well as for its low variance component, which performs well with a relatively low sample size.

Another issue was the preprocessing of attribute variables. As can be seen in Table 8.2, a number of attribute variables were continuous in our study. In machine learning, a discretization is often performed for continuous variables, specially in classifiers designed for discrete variables, such as AODE. We used the *equal frequency* discretization method [55] to split the continuous variables into 10 intervals.

**Feature selection and combination schemes:** The feature selection in the supervised classification paradigm (see Section 2.4 for details) attempts to find a minimum set of attribute variables that can achieve the best performance. The selection of significant features in the problem space can prevent the classifiers from introducing noisy evidence into the training stage. The feature selection can also reduce the number of variables to be considered in the problem space, and can therefore facilitate our understanding of significant variables.

While several techniques have been proposed for the feature selection [78], we used a wrapper method [109]. This method employs a "Best First Search" procedure which evaluates each candidate set of features using the accuracy of an AODE classifier trained with them. The current selection process was similar to the cross validation method described in Section 8.3.3. The 10-fold-cross validation repeated 10 times gives us 100 different data splits with 90% of the samples of the total dataset. In each one of these splits a wrapper feature selection method was run. The final set of features was generated by the features that were selected in at least N% of these 100 splits. We used 50%, 80%, and 90% as the cutoff levels in the feature selection. Again, each one of these cutoff levels was evaluated and the best accuracy was reported.

The feature selection process was independently applied to each one of the seven feature categories defined in Section 8.3.2. Another *combined category* is also evaluated combining the selected features at the same cutoff levels in each one of the feature categories. Again the three nested feature sets were evaluated and the best accuracy was reported.

**Estimating the classification accuracy:** With the aim of providing a robust estimation of the accuracy of a classifier, the set of data was partitioned into two

separated sets. The training set was used to build the classifier and the test set was used to estimate the performance. The K-fold-cross validation method was used to partition the dataset as follows. The dataset $\mathbf{D}$ was divided into $K$ random subsets with the same size $\{D_1, ..., D_K\}$, thus, the validation process was repeated $K$ times. In other words, in step $i$ with $i = 1...K$ a training dataset was defined $T_i = \mathbf{D} \setminus D_i$, subset $D_i$ was used as a test set, and accuracy measurement was based on these. The mean of the $K$ accuracy measures was reported as the final estimated performance of the classifier. In our study, a 10 fold-cross validation was repeated 10 times to measure performance (i.e., based on 100 repeated estimations).

Accuracy of prediction is defined by the portion of correct predictions in the total number of click-through documents. The correct prediction is a sum of true positive and true negative cases (i.e., predicting a relevant document as relevant, and predicting a non-relevant as non-relevant). This prediction accuracy was used to represent the impact of contextual features for relevance modelling. In other words, a stronger contextual feature made it easier for classifiers to model the relevance of documents.

In order to detect statistically significant improvements for the expected probability, 50 % (i.e., Relevant or not), due to the balancing step of the training dataset, we used a corrected paired t-test [134]. This test is stricter than the common t-test and was specifically designed for considering the overlapping between the datasets used.

Because three cutoff levels were used in the feature selection step (Section 8.3.3), the classifier was evaluated with three different feature sets. Thus, three dependent hypothesis were simultaneously tested. To address this issue, we also applied the *Bonferroni correction* [18] to the previous t-test dividing by 3 the statistical level of the test.

In same cases, there were significant increments in classification accuracy in relation to 50%, but these were not meaningful. This is mainly due to the fact that the feature selection stage left an empty or very reduced set of selected features. In many of these cases, classifiers always predicted the same class (i.e., non-relevant) independently from the query-independent feature combinations they received. Thus, they showed a classification accuracy rate higher than 50%. This problem

was detected by observing the true positive and true negative classification rates and discarding those estimations in which any of these two values were lower than 50% (a completely non-informative prediction presents 100% of true positive rate and 0% of true negative rate or viceversa). We will refer to those discarded cases as *degenerated relevance modelling*.

In short, our approach was designed to divide the dataset (click-through documents and subsequent relevance judgements) based on a context. We then trained the classification model with individual sets of data to measure the impact of context on relevance modelling.

Dividing data to measure the impact of a variable is not a new concept in classification. For example, Lachin's approach [116] was based on building these divisions for making predictions through the application of Bayesian decision rules. A stepwise procedure was employed to select a minimum set of variables and their Cartesian product was then used to define each one of the splits of the data. Our approach differed from Lachin's in the following way. In Lachin's approach, each split described a unique class prediction (for instance, always relevant) and the set of variables involved was always the same. When we used a context variable, an independent feature selection process was applied to the data corresponding to its different values. In our approach, in each set of data, the selected set of variables could therefore be different, as well as the way in which they influenced the prediction. This was considered to be more appropriate for measuring the impact of different contextual factors, in comparison to Lachin's approach.

## 8.4 Experiments

This section presents the experimental design of our study. We first provide an overview of the original studies from which the experimental data were extracted. The set of contexts examined is then discussed. Finally, the data on click-through documents and relevance assessments are presented.

### 8.4.1 Overview of original studies

While a new study can be conducted to collect experimental data and measure the impact of context, we nevertheless decided to revisit two former experiments as a preliminary study of the proposed approach. More specifically, two user studies [100; 101], conducted independently, formed the basis of our investigation. An overview of the two original studies is as follows.

**Study 1** The first of the two original studies investigated the effects of the level of document representations on searchers' interaction with a search engine [101]. This study will be referred as to *Study 1* in this research. Four interfaces were devised in Study 1 to vary the level of document representations shown in search results as document surrogates. Study 1 was particularly interested in the effectiveness of textual and visual representations as the additional component in document surrogates. The baseline interface had no additional representation and was based on Google's search result presentation. The second interface augmented the baseline interface with top-ranking sentences (*TRS*) as the additional textual representation. A version of the software originally developed by [178] was used to generate TRS. The third interface augmented the baseline interface with a thumbnail image of documents as the additional visual representation. The final interface combined the TRS and thumbnail image.

24 participants were recruited in the study and each of them performed a search task for four topics using a different order of the four interfaces. Participants were also divided into two subject groups based on their search experience: the more experienced group and the less experienced one. The search experience was established by an entry-questionnaire prior to the experiment. There were therefore two independent variables (level of document representation, and search experience) and one controlled variable (topics) in *Study 1*.

**Study 2** The second of the two original studies investigated the effectiveness of a search result browsing support function [100]. This study will be referred as to *Study 2* in this work. Two interfaces were devised in Study 2. The baseline interface was similar to the one used in Study 1. The experimental interface was designed to offer an independent area called *Workspace* in which users can

group the search results based on the terms appearing in the document surrogate of retrieved documents. The design of the browsing support was inspired by a faceted approach to exploring search results. The interface was devised to offer greater control in the way search results are organised and explored.

As in Study 1, 24 participants were recruited in the study and each performed four search tasks using a different order of the two interfaces. However, the topics were different from Study 1. In addition, two levels of complexity were manipulated for the four search tasks by varying the amount of description given for the task background and relevant information. Thus, there were also two independent variables (presence of the browsing support, and task complexity) and one controlled variable (topics).

Simulated work task framework [19] was used in both studies to facilitate participants' engagement in the simulated tasks. The framework described a task as a type of short scenario. The scenario explained the contexts and motivation of the search with sufficient information on the relevance of pages. Participants were asked to bookmark the documents during the tasks when they perceived relevant information to be found. User interaction with the interfaces was recorded so that all click-through documents and their relevance judgements made by participants were used in the analysis.

### 8.4.2 Contexts and sub-groups

Based on the independent and controlled variables used in the original studies, we formed a set of context to investigate their impact on searchers' relevance assessments. In addition, we devised *Search stage* context which was suggested to be affective in [178]. As a result, six main contextual factors and three interaction factors were formulated, as shown in Table 8.3. The main contexts were Topic, Search experience, Interface I (Document representation), Interface II (Browsing support), Task complexity, and Search stage. The interactions were between Interface II and Task complexity, Interface II and Search stage, and Task complexity and Search stage. As illustrated in Table 8.3, each contextual factor had more than one sub-group. For example, the searchers had greater and lesser degrees of experience in the Search experience context. Two topics in Study 1 were

Table 8.3: Contexts and sub-groups.

| Context | Group | Description | Study |
|---|---|---|---|
| All | all-1 | All documents judged by participants | 1 |
| | all-2 | All documents judged by participants | 2 |
| Topic | topic-1 | Recent change of student populations | 1 |
| | topic-2 | Best Hi-Fi speakers available within a target price | 1 |
| | topic-3 | Dust allergy in workplace | 2 |
| | topic-4 | Music piracy on the Internet | 2 |
| | topic-5 | Petrol price | 2 |
| | topic-6 | Art galleries and museums in Rome | 2 |
| Search experience | more-exp | More experienced searchers | 1 |
| | less-exp | Less experienced searchers | 1 |
| Interface I | sys-rep1 | Baseline system with no additional representation | 1 |
| (Document | sys-rep2 | Experimental system with TRS | 1 |
| representation) | sys-rep3 | Experimental system with thumbnail | 1 |
| | sys-rep4 | Experimental system with TRS and thumbnail | 1 |
| Interface II | sys-brw1 | Baseline system with no browsing support | 2 |
| (Browsing support) | sys-brw2 | Experimental system with browsing support | 2 |
| Task complexity | low-cmp | Low complexity task | 2 |
| | high-cmp | High complexity task | 2 |
| Search stage | stage-1 | First 1/3 of search session | 2 |
| | stage-2 | Second 1/3 of search session | 2 |
| | stage-3 | Last 1/3 of search session | 2 |
| Interface II | brw1-low | Baseline system (sys-brw1) in low complexity tasks | 2 |
| × | brw1-high | Baseline system (sys-brw1) in high complexity tasks | 2 |
| Task complexity | brw2-low | Experimental system (sys-brw2) in low complexity tasks | 2 |
| | brw2-high | Experimental system (sys-brw2) in high complexity tasks | 2 |
| Interface II | brw1-stg1 | Baseline system (sys-brw1) in the first 1/3 of session | 2 |
| × | brw1-stg2 | Baseline system (sys-brw1) in the second 1/3 of session | 2 |
| Search stage | brw1-stg3 | Baseline system (sys-brw1) in the last 1/3 of session | 2 |
| | brw2-stg1 | Experimental system (sys-brw2) in the first 1/3 of session | 2 |
| | brw2-stg2 | Experimental system (sys-brw2) in the second 1/3 of session | 2 |
| | brw2-stg3 | Experimental system (sys-brw2) in the last 1/3 of session | 2 |
| Task complexity | low-stg1 | First 1/3 of search sessions in low complexity tasks | 2 |
| × | low-stg2 | Second 1/3 of search sessions in low complexity tasks | 2 |
| Search stage | low-stg3 | Last 1/3 of search sessions in low complexity tasks | 2 |
| | high-stg1 | First 1/3 of search sessions in high complexity tasks | 2 |
| | high-stg2 | Second 1/3 of search sessions in high complexity tasks | 2 |
| | high-stg3 | Last 1/3 of search sessions in high complexity tasks | 2 |

Table 8.4: Number of relevance judgements and documents.

| | | Judgements | | | Documents | | | |
|---|---|---|---|---|---|---|---|---|
| Context | Group | Relevant | Non-rel | Total | Relevant | Non-rel | Discarded | Total |
| All | all-1 | 154 | 670 | 824 | 102 | 457 | 10 | 569 |
| | all-2 | 528 | 513 | 1041 | 341 | 362 | 34 | 737 |
| Topic | topic-1 | 75 | 138 | 213 | 59 | 122 | 2 | 183 |
| | topic-2 | 56 | 212 | 268 | 30 | 144 | 7 | 181 |
| | topic-3 | 155 | 102 | 257 | 117 | 80 | 6 | 203 |
| | topic-4 | 114 | 109 | 223 | 78 | 90 | 5 | 173 |
| | topic-5 | 104 | 137 | 241 | 62 | 85 | 7 | 154 |
| | topic-6 | 155 | 165 | 320 | 84 | 107 | 16 | 207 |
| Search | more-exp | 81 | 333 | 414 | 65 | 266 | 4 | 335 |
| experience | less-exp | 73 | 337 | 410 | 46 | 239 | 5 | 290 |
| Interface I | sys-rep1 | 52 | 139 | 191 | 41 | 113 | 2 | 156 |
| | sys-rep2 | 33 | 170 | 203 | 29 | 146 | 2 | 177 |
| | sys-rep3 | 25 | 179 | 204 | 22 | 149 | 0 | 171 |
| | sys-rep4 | 44 | 182 | 226 | 35 | 145 | 2 | 182 |
| Interface II | sys-brw1 | 271 | 278 | 549 | 196 | 212 | 18 | 426 |
| | sys-brw2 | 257 | 235 | 492 | 192 | 194 | 8 | 394 |
| Task | low-cmp | 236 | 270 | 506 | 164 | 197 | 13 | 374 |
| complexity | high-cmp | 292 | 243 | 535 | 204 | 189 | 12 | 405 |
| Search | stage-1 | 179 | 164 | 343 | 123 | 121 | 9 | 253 |
| stage | stage-2 | 159 | 159 | 318 | 136 | 138 | 4 | 278 |
| | stage-3 | 190 | 190 | 380 | 155 | 165 | 10 | 330 |
| Interface II | brw1-low | 116 | 143 | 259 | 88 | 114 | 6 | 208 |
| × | brw1-high | 155 | 135 | 290 | 120 | 109 | 9 | 238 |
| Task | brw2-low | 120 | 127 | 247 | 98 | 104 | 2 | 204 |
| complexity | brw2-high | 137 | 108 | 245 | 107 | 94 | 5 | 206 |
| Interface II | brw1-stg1 | 84 | 96 | 180 | 64 | 78 | 4 | 146 |
| × | brw1-stg2 | 87 | 78 | 165 | 79 | 70 | 2 | 151 |
| Search | brw1-stg3 | 100 | 104 | 204 | 89 | 94 | 4 | 187 |
| stage | brw2-stg1 | 95 | 68 | 163 | 75 | 56 | 1 | 132 |
| | brw2-stg2 | 72 | 81 | 153 | 69 | 74 | 0 | 143 |
| | brw2-stg3 | 90 | 86 | 176 | 76 | 78 | 5 | 159 |
| Task | low-stg1 | 68 | 93 | 161 | 53 | 69 | 5 | 127 |
| complexity | low-stg2 | 74 | 86 | 160 | 63 | 71 | 2 | 136 |
| × | low-stg3 | 94 | 91 | 185 | 76 | 83 | 4 | 163 |
| Search | high-stg1 | 111 | 71 | 182 | 77 | 58 | 5 | 140 |
| stage | high-stg2 | 85 | 73 | 158 | 77 | 68 | 2 | 147 |
| | high-stg3 | 96 | 99 | 195 | 85 | 87 | 6 | 178 |

removed from Topic context due to the small size of documents for the classifiers. However, all documents were used for other contexts (i.e., Search experience and Interface I) in Study 1. For a similar reason, search stage context was not used in Study 1.

The number of click-through documents and participants judgements with regard to these are shown in Table 8.4. As can be seen, there was a total of 824 and 1041 relevance judgements made on 569 and 737 different documents in Study 1 and 2, respectively. The size of relevance judgements was more balanced in Study 2 than in Study 1. As discussed in Section 8.3.1, we discarded the documents when there was complete disagreement among participants in relation to judgements. The number of discarded documents was shown in the eighth column in the table. As can be seen, the proportion of discarded documents was small across the contextual groups. The resulting dataset formed the basis of our investigation.

## 8.5 Results

This section presents the results of our experiments. Section 8.5.1 shows the effect of context by looking at the performance of relevancy prediction on contextual relevance. Section 8.5.2 shows the performance of feature categories for individual contexts. Finally, Section 8.5.3 looks at the members of feature categories that were found to be effective at modelling contextual relevance.

### 8.5.1 Impact of context

As discussed in Section 3, we used the prediction accuracy of classifiers as the measure of context impact on searchers relevance assessments. Table 8.5 shows an overview of the context impact. The first column shows the contexts examined in our study. The second column presents a set of groups within each context. The third column shows the relative increase of relevancy prediction accuracy from the expected probability of 50% (i.e., Relevant or not). Lastly, the fourth column is the average increase in each context ("-" values were not considered to compute this mean).

Table 8.5: Overall effect of context.

| Context | Group | Increase (%) | Average |
|---|---|---|---|
| All | all-1 | 7.7 | |
| | all-2 | 6.9 | 7.30 |
| Topic | topic-1 | 14.0 | |
| | topic-2 | 10.8 | |
| | topic-3 | 10.0 | |
| | topic-4 | - | |
| | topic-5 | 15.5 | |
| | topic-6 | 14.8 | 13.02 |
| Search experience | more-exp | 11.9 | |
| | less-exp | 10.4 | 11.15 |
| Interface I | sys-rep1 | 12.1 | |
| (Document representation) | sys-rep2 | 16.6 | |
| | sys-rep3 | - | |
| | sys-rep4 | 12.9 | 13.87 |
| Interface II | sys-brw1 | 6.3 | |
| (Browsing support) | sys-brw2 | 7.6 | 6.95 |
| Task complexity | low-cmp | 7.8 | |
| | high-cmp | 7.8 | 7.8 |
| Search stage | stage-1 | 8.1 | |
| | stage-2 | - | |
| | stage-3 | - | 8.1 |
| Interface II | brw1-low | 11.9 | |
| × | brw1-high | 11.1 | |
| Task complexity | brw2-low | 14.8 | |
| | brw2-high | 13.0 | 12.70 |
| Interface II | brw1-stg1 | - | |
| × | brw1-stg2 | 10.4 | |
| Search | brw1-stg3 | - | |
| stage | brw2-stg1 | 12.4 | |
| | brw2-stg2 | 14.6 | |
| | brw2-stg3 | - | 12.47 |
| Task | low-stg1 | 16.8 | |
| complexity | low-stg2 | 11.9 | |
| × | low-stg3 | - | |
| Search | high-stg1 | 12.1 | |
| stage | high-stg2 | - | |
| | high-stg3 | 12.7 | 13.38 |

'-' indicates that there was not any statistically significant increment in any of the feature categories.

For example, when all contexts were considered without grouping of documents, the classifiers predicted the document relevancy in an average of 57.35% of cases. All the increases presented in Table 8.5 are statistically significant compared to the baseline probability of 50%, except for those cases marked with "-" where there was no statistically significant increment in any of the feature categories. We used the corrected paired t-test [135] with the critical value of $p \leq .05$ in this study, unless otherwise stated.

As can be seen in the average increase, the prediction accuracy was generally higher when the grouping of documents was performed to represent a context, compared with the all-contexts set. This constituted important empirical evidence, supporting the idea that modelling of document relevancy can be improved by exploiting the contextual factors of search environments.

The second finding was that the effect of context varied. For example, *Topic* and *Interface I (Document representation)* were found to constitute strong contexts in our experimental conditions, while *Interface II (Browsing support)*, *Task complexity*, and *Search stage* were found to have a relatively weak effect. Similarly, there are strong groups and weak groups within each context. For instance, Topic 4 does not have any feature category with significant increments, while Topic 3 and 4 appeared to have a weaker effect than other topics in *Topic* context. When we looked at *Interface I* and *Interface II* contexts, the baseline systems (i.e., sys-rep1 and sys-brw1) appeared to have a weaker effect than the experimental systems (i.e., sys-rep2, rep4, and sys-brw2) except for sys-rep3, in which no significant increment was observed. This suggests that it is important to examine the performance of sub-groups when a particular context is exploited for relevancy modelling. One may find a particular subgroup easier to model than the other groups within a context. That is to say, for some subgroups, classifiers are capable of predicting the relevancy of their documents but for the documents of other subgroups, this will not be possible.

The third finding was that the interaction effect of contexts was worth investigating. In our experimental conditions, contexts such as *Browsing support* and *Task complexity* were found to have a weak effect on relevancy modelling. The performance was relatively similar across the sub-groups of the two contexts.

However, the strength of effect increased when they interacted with other contextual factors. For example, group *brw2-low*, showed one of the strongest effects among all context groups. This highlights the importance of examining the dependency of contextual factors. This aspect will be discussed in Section 8.5.2 in greater detail.

### 8.5.2 Context and feature categories

This section presents the relationship between context and eight feature categories used to model contextual relevance. In our experiments, there were seven query-independent document feature categories and one that combined them all (see Section 8.3.3 for details). In other words, the performance of single feature categories indicated whether or not contextual relevance was associated with a particular aspect of documents. The performance of the combined category, on the other hand, indicated whether or not contextual relevance required the range of feature categories for effective modelling (all features independently selected in each of the seven feature categories were considered by the classifier for modelling relevance). Table 8.6 shows the increase in relevancy prediction based on the query-independent document feature categories. The two bottom rows of the table show the average and standard deviation of the increase over the context groups.

One of the aspects we attempted to discover was the robustness of feature categories across different context groups. We were interested in finding whether any feature category worked well across the range of contexts. The results show, however, that the effective category for modelling contextual relevance varies. Indeed, none of the single categories performed well enough to stand out from the crowd.

The lack of consistent performance of single categories seems to lead the combined category (*Comb'd* column) achieving the best performance in many context groups. The average increase supports this. Therefore, the results suggest that combining evidences from different feature categories is the most robust way to model contextual relevance. This is not surprising given that the combined category employed the range of document features to capture the effect caused by

Table 8.6: Performance of feature categories.

| Context | Group | Text | Visual | Vis-tag | Layout | Structure | Other | Oth-tag | Comb'd |
|---|---|---|---|---|---|---|---|---|---|
| All | all-1 | - | - | 5.6 | **7.7** | 4.2 | - | 4 | - |
| | all-2 | **6.9** | 3.8 | **6.2** | 5.1 | - | - | **6.1** | **6.8** |
| Topic | topic-1 | 4.4 | - | 6.2 | 5.4 | **14** | - | **10.9** | 7.6 |
| | topic-2 | 5.2 | **10.2** | - | **10.8** | - | - | - | 4.8 |
| | topic-3 | **10** | - | - | 6.8 | 8.3 | 8.7 | 6.2 | **9.6** |
| | topic-4 | 3 | - | 3.8 | 10 | 8.9 | 2 | 4.9 | 9.1 |
| | topic-5 | 2.6 | **12.2** | **11.7** | 9 | 8.9 | **15.5** | **12.9** | **13.8** |
| | topic-6 | 8.8 | **13.1** | 6.2 | **9.5** | **9.7** | **10.4** | **8.6** | **14.8** |
| Search | more-exp | 3.8 | - | 4.2 | **7.5** | **11.9** | - | **10.3** | **8.6** |
| experience | less-exp | 5.1 | 0.8 | **10.4** | 6 | - | - | - | - |
| Interface I | sys-rep1 | - | **12.1** | - | - | 5.6 | - | - | 5.9 |
| | sys-rep2 | 5.9 | 2.8 | - | 11.1 | **16.6** | 8.3 | 4.5 | **11.5** |
| | sys-rep3 | - | 5.1 | 3.2 | 7.3 | 7.3 | - | - | 7.8 |
| | sys-rep4 | - | 9.8 | **12.9** | 6.4 | - | - | - | 2.7 |
| Interface II | sys-brw1 | - | 4.3 | 4.4 | 1.8 | - | - | **6.3** | 5.9 |
| | sys-brw2 | **7.3** | 6.2 | **7.2** | **7.4** | - | **7.6** | 4.2 | **7.5** |
| Task | low-cmp | 5.3 | 5.3 | 3.8 | 4.4 | 2.5 | - | **7.8** | 4.8 |
| complexity | high-cmp | 3.5 | 4.1 | 5.9 | 1.3 | 1.7 | **6.6** | 1.7 | **7.8** |
| Search | stage-1 | 5 | 5.3 | **8.1** | 3.7 | 2 | - | 4.3 | **7.8** |
| stage | stage-2 | 5.5 | 6.4 | - | 4.9 | - | 4 | 2.6 | 5.0 |
| | stage-3 | 4.8 | - | 1.7 | 2.4 | 4.9 | - | 3.2 | 4.6 |
| Interface II | brw1-low | - | - | 7.2 | 4.9 | 2.9 | **11.9** | 7.4 | 5.6 |
| × | brw1-high | 8 | 3.1 | 6.2 | - | **11.1** | - | **8.5** | **9.7** |
| Task | brw2-low | **11.4** | - | **11.2** | **9.5** | - | **14.8** | 5.9 | **11.3** |
| complexity | brw2-high | 7.4 | **10.5** | - | **9** | 3.2 | 9.7 | **8.6** | **13.0** |
| Interface II | brw1-stg1 | - | - | 9.9 | - | 4.9 | - | 10.2 | 7.2 |
| × | brw1-stg2 | 8.9 | 7.9 | - | - | - | **10.4** | 5.2 | 4.4 |
| Search | brw1-stg3 | - | - | - | 4.8 | 8.5 | 6.7 | 6.3 | 7.0 |
| stage | brw2-stg1 | 4.7 | 9.1 | 7.2 | **11.6** | 4 | - | 6.9 | **12.4** |
| | brw2-stg2 | **12.9** | **12.5** | 5.7 | 5.8 | 8.2 | **11.8** | 2.8 | **14.6** |
| | brw2-stg3 | 1.3 | 10.4 | - | 3.1 | 4.2 | - | 8.7 | 9.6 |
| Task | low-stg1 | **11.2** | - | **16.8** | 4.4 | 5.3 | - | 8.6 | **13.2** |
| complexity | low-stg2 | 8 | - | - | 7.9 | 5.3 | **11.9** | - | **11.9** |
| × | low-stg3 | 7.6 | 2.5 | - | 1.7 | - | - | 6.3 | 7.0 |
| Search | high-stg1 | - | 6.1 | 5.7 | 8.3 | 10.5 | 5.9 | 4.6 | **12.1** |
| stage | high-stg2 | 4.2 | 5.5 | 5.5 | 1.7 | 1.6 | 4.9 | 8.4 | 5.0 |
| | high-stg3 | 8.1 | 8.6 | 8.1 | - | **12.7** | 3 | 3.9 | 5.2 |
| | Mean | 6.14 | 6.03 | 6.77 | 5.73 | 6.66 | 7.56 | 5.61 | 8.45 |
| | SD ($\sigma$) | 2.92 | 3.76 | 3.47 | 2.96 | 4.10 | 4.40 | 2.70 | 3.30 |

**Bold**: Increase is stastistically significant by t-test. Underline: Highest increase in each context group.
"-" indicates there was a *degenerated relevance modelling*

context in document relevancy. However, the results suggest that it is important to consider not only the textual but also the non-textual features for robust modelling of contextual relevance.

Nonetheless, the performance of feature categories helped us to infer the characteristics of contextual relevance, not from a user's point of view (like [10]), but rather from a modelling perspective. While a user's perception of relevance was useful for the design of better user experience, modelling of relevance was essential for development of search models exploiting context. The following are some of the characterisations of the context groups based on the performance of feature categories.

In *Search experience* context, the more experienced and less experienced groups had a very different feature category to model the relevance. While category *Structure* was found to be the most effective in the more experienced group, *Vis-tag* category was the most effective in the less experienced one. This is a sound example with regard to demonstrating that the sub-groups in the same context are closely associated with a different category of document features. Perhaps the less experienced searchers were more likely to make relevance assessments based on the visual effect of web pages than the more experienced ones. However, searchers might not be aware of such behaviour, since this characteristic was inferred by the document features.

In *Interface I* context (Document representation), the relevancy of documents was modelled by different features categories. The baseline system *sys-rep1* was modelled by category *Visual*; in *sys-rep2* (TRS), *structure* stands out as the stronger category; *sys-rep3* (Thubmanil) does not have any feature category that significantly models relevance; and in *sys-rep4* (Thubmanil+TRS), category *Visual-Tag* appears as the relevant one. Unlike the *Search experience* context, we found no pattern about the behaviour of feature categories across the different context groups.

As discussed in Section 8.5.1, the overall effect of *Interface II* (Browsing support) and *Task complexity* contexts was found to be weak in our experimental conditions. However, the effect was strengthened by their interaction. For example, the document relevancy of sys-brw1 (no browsing support) was only modelled by category *Oth-tag* when no task complexity was considered. However, it can

be seen that category *Other* and category *Structure* were more effective with regard to modelling the contextual relevance in the low complexity tasks and high complexity tasks, respectively, for sys-brw1.

On the other hand, sys-brw2 (with browsing support) had several feature categories that were effective for modelling the relevance when no task complexity was considered. The interaction results show that the effect of the browsing support was stronger in the low complexity tasks. And the *vis-tag*, *Other*, *Text* and *Layout* categories were effective for modelling the contextual relevance for sys-brw2. In the high complexity tasks, the *Visual Layout* and *Other-tag* categories were effective. The interaction of two (weak) contexts therefore helped us to elicit the stronger contextual groups based on the distribution of performance increase across the feature categories.

The results of *Search stage* suggest that searchers might be shifting the relevance criterion as the search progresses. The *visual-tag* features were effective for modelling relevance in the first 5 minutes of the search, while it became more difficult for the classifiers to model the relevance in the second stage (stage-2) and the last 5 minutes (stage-3) of the search. The interactions with other contexts show an improvement in prediction performance as was seen in the interaction between *Interface II* and *Task complexity*. For example, *Visual-tag* was effective in the first 5 minutes of the search, but was much more effective in this interval when the search was in low-complexity tasks, while the relevance modelling disappears for high-complexity ones. This explains why it was weakly detected by classifiers in the *Search stage* context: there was a mixture of strong (low-stage1) and ineffective modelling (high-stage1).

This again suggests that there was a interdependence among context groups, which affected people's relevance assessments. However, it became clear from the results that it is not always easy to understand why a certain category worked better than the others in a particular context group.

With respect to the behaviour of the combined category, it can be said that this was the one that showed the best performance. It provides the best averaging classification accuracy and most often reaches the highest accuracy in a context. Although, the single feature category is the one that obtains the best

Table 8.7: Effective variables based on all context groups.

| | Feature selection | | | Average performance increase | | |
|---|---|---|---|---|---|---|
| Rank | Category | Feature | Rate (%) | Category | Feature | Increase (%) |
| 1 | Structure | PageRank (PR) Score | 91.9 | Layout | `p` tag | 6.6 |
| 2 | Layout | `size` att | 91.2 | Layout | `size` att | 6.6 |
| 3 | Layout | `p` tag | 90.7 | Layout | `div` tag | 6.3 |
| 4 | Structure | URL level | 90.3 | Layout | `align` att | 6.3 |
| 5 | Vis-tag | `b` tag | 90.2 | Layout | `br` tag | 6.3 |
| 6 | Structure | HTML link | 89.9 | Vis-tag | `b` tag | 6.2 |
| 7 | Layout | `div` tag | 87.9 | Structure | Outlink | 5.8 |
| 8 | Text | †Digit | 87.2 | Layout | †`li` tag | 5.7 |
| 9 | Layout | `br` tag | 86.5 | Structure | PageRank (PR) Score | 5.7 |
| 10 | Layout | `align` att | 86.5 | Visual | Image area | 5.7 |
| 11 | Structure | Host's PR Score | 86.1 | Visual | `style` tag | 5.6 |
| 12 | Text | †Page length | 85.3 | Structure | URL level | 5.6 |
| 13 | Structure | †Outlink | 85.3 | Oth-tag | `src` att | 5.5 |
| 14 | Visual | Image area | 84.6 | Oth-tag | †`meta` tag | 5.4 |
| 15 | Structure | Non-HTML link | 83.2 | Oth-tag | †`script` tag | 5.3 |
| 16 | Oth-tag | `src` att | 83.0 | Structure | Host's PR Score | 5.3 |
| 17 | Visual | `style` tag | 81.0 | Visual | Disk size of image | 5.2 |
| 18 | Visual | Disk size of image | 80.8 | Vis-tag | `style` att | 5.2 |
| 19 | Structure | †Inlink | 80.4 | Structure | Non-HTML link | 5.2 |
| 20 | Vis-tag | `style` att | 79.6 | Vis-tag | †`H1-6` tag | 5.2 |

†Features that appear in only either of the lists. The HTML tags and attributes (denoted as att), links, and digits are based on their frequency of occurrence in a document.

accuracy and, consequently, the best relevance modelling in a given context (see, for example, *sys-resp1, sys-resp4, high-stage3*).

### 8.5.3 Effectiveness of document features

The previous sections looked at the effects of context and its relationship with the feature categories. This section investigates the individual features that were effective for modelling contextual relevance. The effectiveness of features was measured by the increase in the robustness and performance of prediction accuracy. Robustness was defined as the likelihood of a feature being selected of modelling the relevance of the context groups (see Section 8.3.3 for the feature selection process). In other words, when a feature was selected more frequently within each feature category to model the contextual relevance, it was seen to be more robust. The increase in performance was, on the other hand, the current contribution made by the feature with regard to improving prediction accuracy. This value was computed by averaging the classification accuracy rates listed in Table 8.6 in which the feature was included for relevance modelling, following the

steps detailed in Section 8.3.3. Table 8.7 lists the top 20 features ranked by the increase in robustness and average performance based on all context groups.

From column *Feature selection*, we can see the features that were frequently selected within each feature category. For example, the PageRank score and URL levels were selected in the feature selection process in 90% of cases, and can therefore be seen as the robust features in category *Structure*.

On comparing the lists, many features appear in both of them. This is because there is a correlation between the increases in robustness and performance. However, some features did not appear in the *Average performance increase* column. One example involves the two features in *Text* category. The number of digits in a document and length of the documents were both found to be robust in the category. However, since the *Text* category's overall prediction accuracy was relatively low across the context groups, neither feature was ranked in the top 20 in the list of average performances increases. Instead, the features in *Layout* category (i.e., number of `li` tags) and the *Oth-tag* categories (i.e., number of `meta` and `script` tags) were ranked higher.

In short, the features listed in Table 8.7 were frequently selected to model the contextual relevance in different context groups, as shown by the feature selection rate. This means that, while the performance of feature categories varies across the context groups, the effective features within the individual categories remain consistent. We observed the top ranked features in individual context groups and confirmed that this was the case for most of them. Another implication is that these features were frequently used in the combined category discussed in Section 8.5.2.

## 8.6 Discussion

This chapter presented an approach for measuring the impact of context by looking at the relevance model derived from documents assessed in particular context groups. This section first summarises the main findings of the experimental results of our study. The implications of the findings are then discussed, followed by the limitations of our study.

### 8.6.1 Main findings

As discussed in Section 8.2, one of the objectives of our research was to make a methodological advance in IR in the context area by developing a framework for measuring the impact of context. Our results showed that grouping of click-through documents was a viable way to represent a context and to infer the impact on people's relevance assessments. The framework proposed also enabled us to examine the dependency of contextual factors. The findings of our study were as follows.

The first finding was that query-independent document features can be successfully exploited to model contextual relevance. The document features investigated in our study performed significantly better than the value expected in many context groups. More significantly, the relevance models derived from contextually-partitioned document sets almost always performed better than the models derived from the whole set of documents. In other words, the effect of context was elicited through document grouping, and the document features were capable of quantifying it to model relevance. This supported our approach for measuring the strength of context in people's relevance assessments.

The results showed that topics, search experience, and document representation clearly biased people's relevance assessments, and we therefore found significant contexts in our experimental setting. On the other hand, browsing support, task complexity, and search stage generally exerted a lower level of bias than the first three contexts. However, a stronger effect was observed when these contexts interacted with other factors. The interaction effect was consistent across the weak contexts. This empirically demonstrated that greater knowledge of the context of search environments helped to increase the accuracy of relevance modelling.

Another finding was that no single document feature category showed consistent performance across the context groups. Most categories performed well in some context groups but poorly in the others. As far as average performance was concerned, the difference was small across the feature categories (See Table 8.6). This was somehow disappointing, as we wished to find one or two robust features for modelling relevance in a range of contexts. Instead, we found a relationship

between strength of context and feature categories. More specifically, as the bias of context strengthened, it became more likely that the contextual relevance could be modelled by a single feature category. When the bias was weak, there was a need to employ the combined category.

The last finding was that the performance of features in a given document-feature category was relatively consistent. In other words, a similar set of features was often selected in the feature selection stage and used to model contextual relevance within the same category across context groups. This meant that there was a consistent membership of effective features within individual feature categories.

### 8.6.2 Implications

First, the findings of this study have implications for the design and development of IR applications exploiting context. First and foremost, our results support the benefits of leveraging context in order to improve a system's modelling of relevance [92]. We showed many cases in which the prediction accuracy of document relevancy improved significantly when a contextual factor was available in search environments. Measuring the potential effect of candidate factors will facilitate the development of context-aware search models.

Second, an understanding of individual search environments is essential with regard to eliciting significant context. Our study showed varying levels of impact across the contextual factors. Based on the findings, we speculate that a contextual factor that works consistently across different environments is not likely to be found. Instead, a better strategy appears to involve collecting the user data in a target environment and measuring the impact of candidate contextual factors. This is similar to supervised machine learning techniques, where the training data were supplied to train classifiers. The sampling of search environments can help us find effective contextual factors.

Third, the subgroups of a context may require different features in order to model relevance. We rarely found a case in which the same document feature category worked well across the subgroups of contextual factors. This means that even if a system was designed to leverage a single contextual factor, different features can be used to model individual subgroups for better performance. This

is similar to the findings of [63], who found that different document genres were associated with different work tasks. A related implication is that investigating the dependency of contextual factors appears to be as important as finding significant factors. If there was an important contextual factor in a search environment, and an initial benchmark suggested a weak effect, examining an interaction effect with other candidate factors might show a condition in which the target context factor has a greater impact.

Fourth, the performance of feature categories suggests that the range of document features should be explored in order to achieve robust modelling of contextual relevance. In this study, several feature categories were devised for relevance modelling. Four of these were non-textual features. This has important implications, especially for implicit feedback. While several models have been proposed to capture relevance feedback from users, existing implicit feedback techniques still rely on existing text-based models [e.g., 179]. Our results showed that other document features such as images, layout, and structures constitute promising aspects with regard to modelling contextual relevance. We found many context groups in which these non-textual features outperformed the text-based ones. This also supports the findings of [172], who observed that searchers were influenced by non-textual factors of documents in their relevance assessments.

### 8.6.3 Limitations

Investigation of the effect of context constitutes a vast research area. It would require a number of theoretical and empirical studies to advance our understanding and use of context in IR. This research investigated only a small fraction of such an area. One of the limitations of our study is that the data were collected from user studies performed in controlled environments. The controlled environments enabled us to isolate independent variables as candidate contextual factors. Nonetheless, generalisation of our findings might be limited when different search environments are examined. In particular, we used Google API to retrieve documents in both of the original studies. Collection thereof is not static, and thus, it is possible to update indexes during the time period in which both studies were carried out. No attempt was made to ensure that an identical list of URLs was

retrieved by the same query. Participants in the original studies were mostly university students, although their background varied. Significant context is likely to differ in other populations. We will continue to apply our framework to other data in order to gain a comprehensive understanding of context in search.

Other limitations lie in the concrete implementation of the modelling approach. Preprocessing of the features is a very important and challenging step that can be further improved. Use of only one classifier model, AODE, also limits the relevance modelling capacity. More classifiers could be employed in an attempt to model the complex dependence structures underlying query-independent features. However, multiple hypothesis testing would generate other problems. Likewise, the feature selection method can be further improved and adapted to the peculiarities of this problem. When a non statistically significant modelling was found, it did not imply that the particular feature category was unable to model the contextual relevance. This means that, with the particular modelling employed in this work, no empirical evidence was found to support the possibility of relevance modelling.

## 8.7 Conclusion and future work

Finding relevant information is an activity embedded in multiple layers of context [52; 92]. To facilitate understanding and leveraging of contexts in IR, we proposed an approach that can be used to measure the impact of context on searchers relevance assessments. The approach enabled us to quantify the impact of several contextual factors, provided that these can divide the dataset. Furthermore, we showed that dependence of contextual factors can also be examined by our method. We believe that the approach proposed can be applied to many different environments to provide further insight into the role of context in IR.

There are several issues for future research. One aspect involves investigating other features to model contextual relevance. Interaction features have been shown to be a promising candidate [62]. We are interested in comparing the effectiveness of object features and interaction features on relevance modelling. Another aspect relates to investigation of the impact of contextual factors in multimedia retrieval. People's relevance assessment criteria can differ and therefore

difficult to model in multimedia retrieval, due to the ambiguity of multimedia objects. We are interested in comparing the significant context in textual and multimedia retrieval.

Additionally, the limitations derived from the concrete implementation of the modelling approach detailed in the previous section gives rise to new issues for future work. One of the main aspects thereof involves employment of classification models capable of handling context-specific independencies. Furthermore, better handling of the continuous features, rather than simple discretization, is another important aspect to be considered. In short, we could consider any supervised classification technique that can improve the difficult task of contextual relevance modelling.

# Part V

# Conclusions

# Chapter 9

# Conclusions and Future Works

This last chapter presents the general conclusions of the dissertation. The specific conclusions of each contribution were previously given at the end of each corresponding chapter. The list of publications and future works are also included here.

The whole dissertation has been devoted, as its title indicates, to supervised classification models and their applications to Genomics and Information Retrieval.

Following the introduction, the methodology section considers two different approaches to supervised classification. The first one presents a new semi-Naive Bayes classifier with grouping of cases. This classifier exploits the joining of variables and the grouping of cases in these new compound variables, in order to achieve competitive performance, particularly in terms of quality of class probabilities estimates, while reducing the number of parameters that encoded the conditional distribution. The experimental evaluation showed that this approach demands very low memory resources.

The second methodological advance was related to classification trees. In this case, a Bayesian approach was employed to address some of issues relating to single classification trees and ensembles thereof. We showed how Bayesian techniques can be very useful for solving specific and practical problems relating to these classification models. Concretely, a Bayesian smoothing approach was presented to improve the quality of the probability class estimates produced by classification trees. This only involves some additional effort when learning the

tree, but implies neither more space requirements nor more time in classification time. Moreover, a new random Bayesian split operator has been introduced to build random forests that presented better behaviour in terms of bias-variance error decomposition than its counterpart based on frequentistic or information-based measures.

The application of supervised classification methodologies to real problems was one of the main concerns of this dissertation, as in many cases, the concrete classifier depends of the characteristics of the problem we face. Two different applications were considered. The first one was the classification of the *diffuse large B-cell lymphoma* into two molecular subtypes. This classification was based on gene expression data extracted from tumoral tissues with this disease. A Gaussian Naive Bayes model was employed as the basic classification model. The problem arose with the high dimensionality of this data and the low number of samples. Two different versions of a wrapper approach for feature selection were proposed to address these issues. Both of them performed successfully in this respect when compared with state-of-the-art approaches. Low classification errors and a small set of genes were obtained in both cases.

The other application studied focused on *information retrieval in context*. The problem addressed involved measurement of the strength or the effect that a given contextual factor such as topic in-hand, search experience, task complexity etc. can have in relevance modelling with query-independent features. The use of classification models and feature selection techniques provided suitable tools for addressing this complex problem. Empirical evidence was given of the role played by these contextual factors and its dependency on the relevance assessments made by users in two controlled user studies.

As a general conclusion, this dissertation attempts to contribute to the state of the art of supervised classification models and to application thereof to real problems.

## 9.1 List of Publications

The different studies included in this dissertation have been presented in the following publications (some of them still in revision process):

[1] A. Cano, F. G. Castellano, A.R. Masegosa, and S. Moral, "Application of a selective gaussian naïve Bayes model for diffuse large-b-cell lymphoma classification," in *Proceedings of the Second European Workshop in Probabilistic Graphicals Models*, (Leiden, Holland), pp. 33–40, 2004.

[2] A. Cano, J. G. Castellano, A. R. Masegosa, and S. Moral, "Selective gaussian naïve Bayes model for diffuse large-b-cell lymphoma classification: Some improvements in preprocessing and variable elimination," in *ECSQARU* (L. Godo, ed.), vol. 3571 of *Lecture Notes in Computer Science*, pp. 908–920, Springer, 2005.

[3] A. Cano, F. G. Castellano, A.R. Masegosa, and S. Moral, "Aplicación de un modelo naive Bayes gaussiano con selección de variables al análisis de datos de epxresión genética," in *VI Jornadas de Transferencia Tecnologíca de Inteligencia Artificial, TTIA 2005 (AEPIA). I CONGRESO ESPAÑOL DE INFORMÁTICA.*, (Granada, España), 2005.

[4] A. R. Masegosa, H. Joho, and J. M. Jose, "Identifying Features for Relevance Web Pages Prediction," in First International Workshop on Adaptive Information Retrieval. Glasgow, 2006.

[5] A. R. Masegosa, H. Joho, and J. M. Jose, "Evaluating query-independent object features for relevancy prediction," in *ECIR* (G. Amati, C. Carpineto, and G. Romano, eds.), vol. 4425 of *Lecture Notes in Computer Science*, pp. 283–294, Springer, 2007.

[6] A. R. Masegosa, H. Joho, and J. M. Jose, "Effects of highly agreed documents in relevancy prediction," in *SIGIR* (W. Kraaij, A. P. de Vries, C. L. A. Clarke, N. Fuhr, and N. Kando, eds.), pp. 883–884, ACM, 2007.

[7] A. M. Andrés Cano and S. Moral, "A Bayesian approach to estimate probabilities in classification trees," in *Proceedings of the Fourth European Workshop on Probabilistic Graphical Models* (M. Jaeger and T. D. Nielsen, eds.), pp. 49–56, 2008.

[8] J. Abellán, A. Cano, A. R. Masegosa, and S. Moral, "A semi-naive bayes classifier with grouping of cases," in *ECSQARU* (K. Mellouli, ed.), vol. 4724 of *Lecture Notes in Computer Science*, pp. 477–488, Springer, 2007.

[9] A. Cano, A. R. Masegosa, and S. Moral, "A Bayesian random split to build ensembles of classification trees," to appear in *Ecsqaru*, 2009.

The following references are publications (some of them still in the revision process) in which I have collaborated but not as the main author. Although the subjects they deal with are not included in this dissertation, my work in these publications has also contributed to my training in supervised classification models and their applications.

[10] A. Cano, J. G. Castellano, A. R. Masegosa, and S. Moral, "Methods to determine the branching attribute in Bayesian multinets classifiers," in *ECSQARU* (L. Godo, ed.), vol. 3571 of *Lecture Notes in Computer Science*, pp. 932–943, Springer, 2005.

[11] A. Cano, F. G. Castellano, A. Masegosa, and S. Moral, "Uso de redes bayesianas en el análisis de datos de microarrays de ADN," in *VI Jornadas de Transferencia Tecnologíca de Inteligencia Artificial, TTIA 2005 (AEPIA). I CONGRESO ESPAÑOL DE INFORMÁTICA.*, (Granada, España), 2005.

[12] A. E. Rodrigo, A. Cañas, A. Masegosa, and D. C. J. Álvarez, "Detección de rostros y mejora de la calidad de imagen en fotografías de tipo carné enviadas a una plataforma web," in *Simposio de Inteligencia Computacional, SICO 2005 (IEEE Computational Intelligence Society, SC). I CONGRESO ESPAÑOL DE INFORMÁTICA.*, (Granada, España), 2005.

[13] J. Abellán, S. Moral, M. Gómez, and A. R. Masegosa, "Varying parameter in classification based on imprecise probabilities," in *SMPS* (J. Lawry, E. Miranda, A. Bugarín, S. Li, M. A. Gil, P. Grzegorzewski, and O. Hryniewicz, eds.), vol. 37 of *Advances in Soft Computing*, pp. 231–239, Springer, 2006.

[14] J. Abellán and A. R. Masegosa, "Combining decision trees based on imprecise probabilities and uncertainty measures," in *ECSQARU* (K. Mellouli, ed.), vol. 4724 of *Lecture Notes in Computer Science*, pp. 512–523, Springer, 2007.

[15] J. Abellán and A. R. Masegosa, "Split criterions for variable selection using decision trees," in *ECSQARU* (K. Mellouli, ed.), vol. 4724 of *Lecture Notes in Computer Science*, pp. 489–500, Springer, 2007.

[16] J. Abellán and A. Masegosa, "Requirements for total uncertainty measures in dempster-shafer theory of evidence," *International Journal of General Systems*, vol. 37, pp. 733–747, 12 2008.

[17] J. Abellán, A. Masegosa, and M. Gómez, "A filter-wrapper method to select variables for the naive bayes classifier based on credal decision trees," *Submitted to IJUFKS*, 2009.

[18] J. Abellán and A. Masegosa, "Ensembling credal decision trees," *Summited to EJOR*, 2009.

[19] J. Abellán and A. Masegosa, "An experimental study about simple decision trees for bagging ensemble on datasets with classification noise", to appear in *Ecsqaru* 2009.

## 9.2 Future Work

This last section attempts to provide a general overview of the previous specific comments, regarding future research, in each chapter.

Undoubtedly, there are many unanswered questions in the different studies presented in this dissertation. Firstly, in the semi-Naive Bayes with grouping of cases presented in Chapter 3, we plan to work on the application of this approach, involving grouping of cases, to other kinds of classifiers such as AODE (Section 2.2.6) where we expect to achieve the same important reduction in the number of parameters of this model without further reduction in its classification performance.

In the section dealing with the Bayesian account of classification trees, an important part of the research has been left for the future. The employment of a prior distribution of the parameters different from the uniform distribution constituted a key step with regard to improving the quality of probability estimates in the approach presented in Section 4.3. Some of the future work related to this approach could be based on the definition of different prior distributions of the parameters of the tree model in order to make good use of the flexibility that Bayesian approaches offer in this issue. In the part related with ensembles of classification trees, we plan to apply a more thorough Bayesian method to this problem. More exactly, we attempt to apply a Markov chain Monte Carlo method to sample the different classification trees of the ensemble.

In the part of the dissertation related to practical applications, there is also much room for further research. The application of the proposed classifiers to other Genomics problems different for *diffuse large B-Cell lymphoma* classification will provide an additional evaluation of their behavior. Another line of future research could involve the application of Bayesian methodologies in order to integrate prior knowledge that can be extracted from genetic research in this field. For example, addressing the gene selection problem with a Bayesian approach and defining a prior distribution of the genes that are known to have an effect on the diseases analyzed. The utilization of classification models different from the Gaussian naive Bayes is also another important point that needs to be further investigated.

And, finally, the part devoted to *information retrieval in context* also presents very interesting points for further research. There are many open possibilities for the refinement and improvement of the proposed methodology. We are very interested in the employment of classifiers capable of encoding probability distributions with context-specific dependencies which, as we showed, are present in this kind of data. Moreover, the definition of other different features for relevance modelling, as well as the application of this methodology to other related problems in multimedia IR or web page classification, involves other lines for future research.

# Appendixes

## Appendix of Chapter 3

Table 1: Number of Kilobytes of memory needed to define the classification models (Full expanded Table 3.9 of Chapter 3)

| Dataset | NB | SNB-G | WNB | J48 | WSNB | TAN | AODE | HNB | WAODE |
|---|---|---|---|---|---|---|---|---|---|
| anneal | 6.2 | 3.7 | 2.8 | 1.8 | - | 20.9 | 814.5 | 826.4 | 814.8 |
| audiology | 33.4 | 21.8 | 9.4 | 5.6 | 12.3 | 84.7 | 5927.4 | 5965.7 | 5928.0 |
| autos | 6.2 | 3.5 | 2.4 | 2.4 | 37.5 | 26.7 | 693.1 | 698.4 | 693.3 |
| balance-scale | 0.3 | 0.2 | 0.2 | 1.0 | 0.3 | 0.6 | 2.9 | 3.1 | 2.9 |
| wisconsin-cancer | 0.5 | 0.2 | 0.4 | 0.2 | 0.5 | 1.4 | 14.3 | 15.0 | 14.3 |
| horse-colic | 1.0 | 0.4 | 0.2 | 0.1 | 0.9 | 2.9 | 67.6 | 71.7 | 67.8 |
| credit-rating | 0.9 | 0.4 | 0.4 | 0.4 | 1.2 | 2.9 | 47.1 | 49.1 | 47.2 |
| german-credit | 1.0 | 0.5 | 0.4 | 1.4 | 1.2 | 3.3 | 70.1 | 73.6 | 70.3 |
| pima-diabetes | 0.3 | 0.2 | 0.2 | 0.3 | 0.4 | 0.6 | 5.7 | 6.3 | 5.8 |
| Glass | 1.6 | 1.5 | 1.3 | 1.3 | 3.1 | 4.7 | 46.6 | 47.4 | 46.7 |
| cleveland-disease | 0.5 | 0.3 | 0.2 | 0.4 | 0.5 | 1.1 | 15.1 | 16.6 | 15.2 |
| hungarian-disease | 1.3 | 0.7 | 0.7 | 0.2 | 1.6 | 2.9 | 40.2 | 41.7 | 40.3 |
| heart-statlog | 0.4 | 0.2 | 0.2 | 0.3 | 0.2 | 0.6 | 8.9 | 10.4 | 9.0 |
| hepatitis | 0.6 | 0.4 | 0.2 | 0.1 | 0.3 | 1.1 | 22.1 | 25.2 | 22.2 |
| hypothyroid | 2.2 | 0.8 | 0.8 | 0.5 | - | 5.2 | 156.6 | 163.7 | 156.9 |
| ionosphere | 2.2 | 0.3 | 0.5 | 0.2 | 1.2 | 8.5 | 297.1 | 306.7 | 297.4 |
| iris | 0.3 | 0.3 | 0.2 | 0.1 | 0.2 | 0.9 | 4.6 | 4.8 | 4.7 |
| kr-vs-kp | 1.2 | 0.3 | 0.2 | 0.5 | - | 2.4 | 89.9 | 100.6 | 90.2 |
| labor | 0.6 | 0.3 | 0.1 | 0.1 | 0.1 | 1.2 | 19.7 | 22.0 | 19.9 |
| lymphography | 1.7 | 1.3 | 0.9 | 0.5 | 0.7 | 4.8 | 91.0 | 93.8 | 91.1 |
| mushroom | 2.0 | 0.2 | 0.5 | 0.4 | 83.4 | 11.0 | 251.9 | 256.0 | 252.0 |
| primary-tumor | 10.1 | 8.6 | 8.5 | 7.5 | 10.0 | 33.3 | 599.1 | 601.6 | 599.2 |
| segment | 9.6 | 5.6 | 3.3 | 2.3 | - | 84.1 | 1682.5 | 1685.6 | 1682.6 |
| sick | 1.0 | 0.4 | 0.3 | 0.4 | - | 2.0 | 59.7 | 66.7 | 59.9 |
| sonar | 1.3 | 0.4 | 0.2 | 0.2 | 1.0 | 1.8 | 107.5 | 136.6 | 108.0 |
| soybean | 17.7 | 17.1 | 10.8 | 9.1 | 28.1 | 58.5 | 2107.7 | 2117.8 | 2108.0 |
| splice | 6.8 | 4.5 | 2.6 | 4.1 | - | 32.3 | 1967.7 | 1996.8 | 1968.2 |
| car | 0.8 | 0.4 | 0.7 | 3.8 | 1.0 | 2.8 | 19.5 | 19.9 | 19.6 |
| vehicle | 2.3 | 0.9 | 1.1 | 2.2 | 25.7 | 9.0 | 171.6 | 174.4 | 171.7 |
| vote | 0.5 | 0.2 | 0.1 | 0.1 | 0.1 | 1.1 | 18.1 | 20.3 | 18.2 |
| vowel | 4.5 | 5.5 | 4.1 | 8.0 | 463.8 | 19.3 | 232.0 | 233.1 | 232.1 |
| waveform | 3.0 | 0.5 | 1.6 | 6.9 | - | 9.4 | 386.0 | 399.1 | 386.3 |
| zoo | 7.7 | 2.5 | 4.5 | 0.5 | 2.1 | 60.4 | 1086.3 | 1088.8 | 1086.5 |
| Average | 3.9 | 2.5 | 1.8 | 1.9 | 26.1 | 15.2 | 518.9 | 525.4 | 519.1 |
| Desv | 6.4 | 4.8 | 2.7 | 2.6 | 80.0 | 23.3 | 1108.3 | 1114.7 | 1108.4 |
| Minimum | 0.3 | 0.2 | 0.1 | 0.1 | 0.2 | 0.6 | 2.9 | 3.1 | 2.9 |
| Maximum | 33.4 | 21.8 | 10.8 | 9.1 | 463.8 | 84.7 | 5927.4 | 5965.7 | 5928.0 |

* Data sets at WSNB column with "-" symbol indicates a memory overflow at training time.

Table 2: Memory space ratio respect to SNB-G (Full expanded Table 3.10 of Chapter 3)

| DataSet | NB | WNB | J48 | WSNB | TAN | AODE | HNB | WAODE |
|---|---|---|---|---|---|---|---|---|
| anneal | 1.6 | 0.8 | 0.5 | - | 5.6 | 217.4 | 220.5 | 217.5 |
| audiology | 1.5 | 0.4 | 0.3 | 0.6 | 3.9 | 271.5 | 273.2 | 271.5 |
| autos | 1.8 | 0.7 | 0.7 | 10.8 | 7.7 | 200.5 | 202.1 | 200.6 |
| balance-scale | 1.7 | 1.4 | 6.4 | 1.8 | 3.8 | 18.8 | 20.1 | 19.1 |
| wisconsin-cancer | 2.4 | 2.0 | 1.0 | 2.7 | 7.2 | 72.5 | 76.5 | 72.9 |
| horse-colic | 2.6 | 0.5 | 0.2 | 2.4 | 7.5 | 172.4 | 182.9 | 172.8 |
| credit-rating | 2.1 | 1.1 | 0.9 | 2.9 | 7.4 | 117.9 | 122.9 | 118.2 |
| german-credit | 2.2 | 0.9 | 3.0 | 2.6 | 7.1 | 149.9 | 157.3 | 150.2 |
| pima-diabetes | 1.6 | 1.1 | 1.9 | 2.0 | 3.5 | 31.1 | 34.6 | 31.5 |
| Glass | 1.1 | 0.9 | 0.9 | 2.2 | 3.2 | 31.9 | 32.4 | 32.0 |
| cleveland-diseas | 1.8 | 0.9 | 1.5 | 1.9 | 4.1 | 57.3 | 63.1 | 57.7 |
| hungarian-diseas | 1.8 | 1.0 | 0.4 | 2.3 | 4.1 | 57.2 | 59.4 | 57.4 |
| heart-statlog | 1.7 | 0.7 | 1.3 | 1.1 | 2.9 | 39.9 | 46.9 | 40.4 |
| hepatitis | 1.4 | 0.5 | 0.4 | 0.8 | 2.7 | 53.7 | 61.3 | 54.1 |
| hypothyroid | 2.7 | 0.9 | 0.6 | - | 6.4 | 192.0 | 200.6 | 192.3 |
| ionosphere | 6.3 | 1.4 | 0.6 | 3.6 | 24.8 | 867.9 | 895.9 | 868.7 |
| iris | 1.0 | 0.5 | 0.3 | 0.5 | 2.8 | 14.2 | 14.8 | 14.4 |
| kr-vs-kp | 3.8 | 0.6 | 1.4 | - | 7.7 | 284.4 | 318.2 | 285.3 |
| labor | 1.7 | 0.4 | 0.2 | 0.4 | 3.5 | 60.0 | 66.8 | 60.4 |
| lymphography | 1.3 | 0.7 | 0.4 | 0.6 | 3.8 | 72.6 | 74.8 | 72.7 |
| mushroom | 9.5 | 2.4 | 1.9 | 399.1 | 52.4 | 1205.6 | 1225.4 | 1206.5 |
| primary-tumor | 1.2 | 1.0 | 0.9 | 1.2 | 3.9 | 69.8 | 70.1 | 69.9 |
| segment | 1.7 | 0.6 | 0.4 | - | 15.0 | 299.5 | 300.1 | 299.6 |
| sick | 2.6 | 0.8 | 1.2 | - | 5.4 | 160.9 | 179.9 | 161.6 |
| sonar | 3.4 | 0.6 | 0.6 | 2.5 | 4.6 | 279.8 | 355.4 | 281.0 |
| soybean | 1.0 | 0.6 | 0.5 | 1.6 | 3.4 | 123.2 | 123.8 | 123.3 |
| splice | 1.5 | 0.6 | 0.9 | - | 7.2 | 436.9 | 443.4 | 437.0 |
| car | 1.9 | 1.8 | 9.4 | 2.5 | 6.9 | 48.2 | 49.1 | 48.3 |
| vehicle | 2.5 | 1.2 | 2.3 | 27.7 | 9.7 | 185.0 | 188.0 | 185.2 |
| vote | 2.5 | 0.6 | 0.4 | 0.3 | 5.0 | 84.9 | 95.5 | 85.5 |
| vowel | 0.8 | 0.7 | 1.4 | 83.7 | 3.5 | 41.9 | 42.1 | 41.9 |
| waveform | 6.3 | 3.4 | 14.7 | - | 19.9 | 814.1 | 841.8 | 814.7 |
| zoo | 3.0 | 1.8 | 0.2 | 0.8 | 23.8 | 427.9 | 428.9 | 428.0 |
| Mean | 2.4 | 1.0 | 1.7 | 21.5 | 8.5 | 217.0 | 226.3 | 217.3 |
| Deviation | 1.8 | 0.6 | 2.9 | 69.1 | 9.6 | 265.2 | 271.5 | 265.4 |
| Minimum | 0.8 | 0.4 | 0.2 | 0.4 | 2.7 | 14.2 | 14.8 | 14.4 |
| Maximum | 9.5 | 3.4 | 14.7 | 399.1 | 52.4 | 1205.6 | 1225.4 | 1206.5 |

* Data sets at WSNB column with "-" symbol indicates a memory overflow at training time.

Table 3: Accuracy Performance (Full expanded Table 3.11 of Chapter 3)

| Dataset | SNB-G | NB | AODE | TAN | WAODE | HNB | WNB | J48 | WSNB * |
|---|---|---|---|---|---|---|---|---|---|
| anneal | 98.13 | 95.95 • | 97.88• | 98.32 ○ | 98.63 ○ | 98.20 ○ | 97.95 • | 98.57 ○ | - |
| audiology | 72.98 | 72.64 • | 72.73• | 72.86 • | 77.05 ○ | 73.94 ○ | 76.09 ○ | 76.73 ○ | 73.35 ○ |
| autos | 71.58 | 65.17 • | 74.76○ | 78.40 ○ | 81.08 ○ | 80.83 ○ | 70.26 • | 80.79 ○ | 71.40 • |
| balance-scale | 73.08 | 71.56 • | 69.96• | 71.36 • | 70.06 • | 69.67 • | 71.55 • | 77.82 ○ | 72.33 • |
| wisconsin-cancer | 95.85 | 97.20 ○ | 97.05○ | 96.47 ○ | 97.00 ○ | 96.20 ○ | 96.81 ○ | 95.01 • | 96.38 ○ |
| horse-colic | 80.71 | 79.54 • | 82.45○ | 82.07 ○ | 81.74 ○ | 81.63 ○ | 84.16 ○ | 85.13 ○ | 83.45 ○ |
| credit-rating | 85.01 | 86.22 ○ | 86.67○ | 85.71 ○ | 86.17 ○ | 84.88 • | 85.48 ○ | 85.68 ○ | 85.78 ○ |
| german-credit | 74.59 | 75.04 ○ | 75.83○ | 74.25 • | 75.72 ○ | 75.65 ○ | 73.53 • | 71.13 • | 72.99 • |
| pima-diabetes | 74.66 | 75.26 ○ | 75.70○ | 75.56 ○ | 75.61 ○ | 74.57 • | 75.19 ○ | 74.49 • | 74.45 • |
| Glass | 71.87 | 71.94 ○ | 74.53○ | 73.11 ○ | 73.26 ○ | 73.77 ○ | 71.97 ○ | 67.63 • | 70.15 • |
| cleveland-disease | 82.64 | 83.47 ○ | 82.87○ | 81.85 • | 82.61 • | 81.95 • | 79.77 • | 77.17 • | 80.86 • |
| hungarian-disease | 83.00 | 84.20 ○ | 84.33○ | 84.13 ○ | 85.28 ○ | 84.87 ○ | 81.69 • | 80.16 • | 80.70 • |
| heart-statlog | 83.07 | 82.56 • | 82.70• | 82.48 • | 82.07 • | 82.74 • | 81.44 • | 78.15 • | 79.63 • |
| hepatitis | 83.85 | 84.34 ○ | 85.36○ | 84.01 ○ | 84.52 ○ | 85.55 ○ | 82.47 • | 79.22 • | 80.93 • |
| hypothyroid | 99.03 | 98.19 • | 98.53• | 99.23 ○ | 99.14 ○ | 99.06 ○ | 98.83 • | 99.54 ○ | - |
| ionosphere | 89.09 | 89.40 ○ | 91.09○ | 91.83 ○ | 92.40 ○ | 91.48 ○ | 90.77 ○ | 89.74 ○ | 90.15 ○ |
| iris | 93.33 | 93.33 ○ | 93.07• | 93.80 ○ | 92.93 • | 92.07 • | 93.00 • | 94.73 ○ | 93.40 ○ |
| kr-vs-kp | 92.35 | 87.79 • | 91.03• | 92.05 • | 94.18 ○ | 92.35 ○ | 94.35 ○ | 99.44 ○ | - |
| labor | 90.10 | 88.57 • | 88.43• | 90.40 ○ | 91.57 ○ | 90.83 ○ | 87.83 • | 78.60 • | 87.67 • |
| lymphography | 85.58 | 85.10 • | 86.86○ | 86.65 ○ | 88.22 ○ | 85.57 • | 81.42 • | 75.84 • | 79.69 • |
| mushroom | 99.96 | 95.76 • | 99.95• | 99.99 ○ | 99.98 ○ | 99.96 ○ | 99.63 • | 100.00 ○ | 100.00 ○ |
| primary-tumor | 46.94 | 49.71 ○ | 49.77○ | 46.76 • | 47.94 ○ | 47.85 ○ | 44.01 • | 41.21 • | 39.79 • |
| segment | 94.60 | 91.15 • | 95.07○ | 95.23 ○ | 96.59 ○ | 96.47 ○ | 93.61 • | 96.79 ○ | - |
| sick | 97.48 | 97.12 • | 97.33• | 97.40 • | 97.72 ○ | 97.54 ○ | 97.66 ○ | 98.72 ○ | - |
| sonar | 75.25 | 76.71 ○ | 77.05○ | 76.51 ○ | 77.24 ○ | 76.13 ○ | 72.34 • | 73.61 • | 71.39 • |
| soybean | 93.84 | 92.94 • | 93.21• | 95.23 ○ | 94.33 ○ | 94.67 ○ | 92.46 • | 90.82 • | 92.55 • |
| splice | 94.07 | 95.42 ○ | 96.12○ | 95.39 ○ | 96.36 ○ | 96.13 ○ | 95.35 ○ | 94.08 ○ | - |
| car | 97.69 | 85.46 • | 91.41• | 94.44 • | 90.94 • | 93.01 • | 85.26 • | 92.22 • | 76.54 • |
| vehicle | 69.19 | 61.06 • | 70.32○ | 71.22 ○ | 70.89 ○ | 70.62 ○ | 63.28 • | 72.28 ○ | 67.32 • |
| vote | 95.08 | 90.02 • | 94.28• | 94.69 • | 94.36 • | 94.32 • | 96.18 ○ | 96.57 ○ | 95.59 ○ |
| vowel | 67.07 | 61.99 • | 71.47○ | 68.73 ○ | 75.26 ○ | 74.00 ○ | 62.07 • | 79.82 ○ | 72.26 ○ |
| waveform | 78.97 | 79.97 ○ | 85.01○ | 81.49 ○ | 85.05 ○ | 84.87 ○ | 81.40 ○ | 75.25 • | - |
| zoo | 91.05 | 93.21 ○ | 94.66○ | 92.69 ○ | 98.10 ○ | 97.11 ○ | 91.12 ○ | 92.61 ○ | 88.39 • |
| Average | 83.89 | 82.67 | 84.72 | 84.54 | 85.47 | 84.96 | 83.30 | 83.93 | 79.60 |
| H/L | | 15/18 | 20/13 | 23/10 | 27/6 | 24/9 | 13/20 | 18/15 | 9/17 |

○, • indicates a higher or a lower mean respect to the mean of SNB-G
H/L number of data sets with ○ or with • respectively
* Data sets at WSNB column with "-" symbol indicates a memory overflow at training time.

Table 4: Log-likelihood Performance (Full expanded Table 3.11 of Chapter 3)

| Dataset | SNB-G | NB | AODE | TAN | WAODE | HNB | WNB | J48 | WSNB * |
|---|---|---|---|---|---|---|---|---|---|
| anneal | -0.10 | -0.18 ○ | -0.12 ○ | -0.11 ○ | -0.10 ○ | -0.10 ● | -0.14 ○ | -0.19 ○ | - |
| audiology | -2.48 | -4.07 ○ | -4.00 ○ | -4.26 ○ | -3.22 ○ | -4.10 ○ | -1.48 ● | -2.07 ● | -1.64 ● |
| autos | -1.69 | -2.90 ○ | -1.38 ● | -1.48 ● | -1.51 ● | -1.34 ● | -1.42 ● | -1.41 ● | -1.29 ● |
| balance-scale | -0.75 | -0.87 ○ | -0.83 ○ | -0.84 ○ | -0.83 ○ | -0.81 ○ | -1.03 ○ | -0.93 ○ | -1.00 ○ |
| wisconsin-cancer | -0.22 | -0.43 ○ | -0.17 ● | -0.19 ● | -0.18 ● | -0.17 ● | -0.39 ○ | -0.25 ○ | -0.20 ● |
| horse-colic | -0.77 | -1.18 ○ | -0.71 ● | -0.83 ○ | -0.72 ● | -0.72 ● | -0.65 ● | -0.60 ● | -0.62 ● |
| credit-rating | -0.53 | -0.65 ○ | -0.54 ○ | -0.53 ○ | -0.54 ○ | -0.54 ○ | -0.56 ○ | -0.53 ○ | -0.52 ● |
| german-credit | -0.78 | -0.77 ● | -0.74 ● | -0.79 ○ | -0.75 ● | -0.74 ● | -0.77 ● | -0.86 ○ | -0.78 ● |
| pima-diabetes | -0.74 | -0.78 ○ | -0.74 ○ | -0.73 ● | -0.74 ○ | -0.74 ○ | -0.75 ○ | -0.81 ○ | -0.75 ○ |
| Glass | -1.29 | -1.32 ○ | -1.13 ● | -1.29 ○ | -1.18 ● | -1.20 ● | -1.32 ○ | -1.63 ○ | -1.24 ● |
| cleveland-disease | -0.60 | -0.67 ○ | -0.58 ● | -0.58 ● | -0.58 ● | -0.57 ● | -0.67 ○ | -0.77 ○ | -0.62 ○ |
| hungarian-disease | -0.64 | -0.63 ● | -0.56 ● | -0.54 ● | -0.52 ● | -0.50 ● | -0.69 ○ | -0.79 ○ | -0.67 ○ |
| heart-statlog | -0.58 | -0.68 ○ | -0.61 ○ | -0.58 ○ | -0.60 ○ | -0.58 ● | -0.65 ○ | -0.76 ○ | -0.68 ○ |
| hepatitis | -0.65 | -0.80 ○ | -0.62 ● | -0.66 ○ | -0.61 ● | -0.58 ● | -0.65 ○ | -0.70 ○ | -0.65 ○ |
| hypothyroid | -0.06 | -0.08 ○ | -0.07 ○ | -0.05 ● | -0.05 ● | -0.05 ● | -0.07 ○ | -0.03 ● | - |
| ionosphere | -0.71 | -2.34 ○ | -0.88 ○ | -0.93 ○ | -0.76 ○ | -0.81 ○ | -0.62 ● | -0.45 ● | -0.47 ● |
| iris | -0.31 | -0.32 ○ | -0.26 ● | -0.30 ● | -0.29 ● | -0.35 ○ | -0.35 ○ | -0.31 ● | -0.35 ○ |
| kr-vs-kp | -0.26 | -0.42 ○ | -0.35 ○ | -0.27 ○ | -0.29 ○ | -0.31 ○ | -0.44 ○ | -0.05 ● | - |
| labor | -0.34 | -0.39 ○ | -0.41 ○ | -0.37 ○ | -0.34 ○ | -0.34 ● | -0.55 ○ | -0.84 ○ | -0.54 ○ |
| lymphography | -0.59 | -0.62 ○ | -0.54 ● | -0.57 ● | -0.53 ● | -0.57 ● | -0.70 ○ | -0.98 ○ | -0.84 ○ |
| mushroom | -0.00 | -0.17 ○ | -0.00 ○ | -0.00 ● | -0.00 ● | -0.00 ○ | -0.04 ○ | -0.00 ○ | -0.02 ○ |
| primary-tumor | -2.72 | -2.67 ● | -2.62 ● | -2.97 ○ | -2.81 ○ | -2.81 ○ | -2.80 ○ | -3.35 ○ | -2.94 ○ |
| segment | -0.27 | -0.75 ○ | -0.26 ● | -0.27 ○ | -0.19 ● | -0.17 ● | -0.29 ○ | -0.26 ● | - |
| sick | -0.14 | -0.16 ○ | -0.14 ● | -0.13 ● | -0.12 ● | -0.13 ● | -0.13 ● | -0.07 ● | - |
| sonar | -0.80 | -1.21 ○ | -0.97 ○ | -0.78 ● | -0.85 ○ | -0.77 ● | -0.93 ○ | -1.06 ○ | -0.81 ○ |
| soybean | -0.33 | -0.54 ○ | -0.35 ○ | -0.17 ● | -0.25 ● | -0.19 ● | -0.36 ○ | -1.31 ○ | -0.36 ○ |
| splice | -0.28 | -0.21 ● | -0.17 ● | -0.21 ● | -0.17 ● | -0.17 ● | -0.21 ● | -0.33 ○ | - |
| car | -0.07 | -0.48 ○ | -0.43 ○ | -0.28 ○ | -0.39 ○ | -0.33 ○ | -0.48 ○ | -0.41 ○ | -0.91 ○ |
| vehicle | -1.01 | -2.89 ○ | -0.96 ● | -0.91 ● | -0.91 ● | -0.89 ● | -1.37 ○ | -1.01 ● | -1.07 ○ |
| vote | -0.24 | -0.89 ○ | -0.21 ● | -0.26 ○ | -0.23 ● | -0.25 ○ | -0.20 ● | -0.17 ● | -0.25 ○ |
| vowel | -1.29 | -1.46 ○ | -1.13 ● | -1.22 ● | -0.96 ● | -1.01 ● | -1.47 ○ | -1.53 ○ | -2.13 ○ |
| waveform | -1.03 | -1.10 ○ | -0.47 ● | -0.61 ● | -0.47 ● | -0.49 ● | -0.75 ● | -1.04 ○ | - |
| zoo | -0.20 | -0.18 ● | -0.15 ● | -0.35 ○ | -0.11 ● | -0.16 ● | -0.42 ○ | -0.69 ○ | -0.80 ○ |
| Average | -0.69 | -0.99 | -0.70 | -0.74 | -0.67 | -0.69 | -0.71 | -0.79 | -0.85 |
| H/L | | 5/28 | 19/14 | 16/17 | 21/12 | 22/11 | 9/24 | 11/22 | 8/18 |

○, ● indicates a higher or a lower mean respect to the mean of SNB-G
H/L number of data sets with ○ or with ● respectively
* Data sets at WSNB column with "-" symbol indicates a memory overflow at training time.

# Appendix of Chapter 4

Table 5: Detailed Accuracy Rate (Full expanded Table 4.2, 4.3 and 4.4 of Chapter 4)

| Dataset | $\star C4.5_\rho$ | $\beta_{S=1}$ | $\beta_{S=2}$ | $\beta_{S=\|C\|}$ | $\widehat{\beta}_{S=1}$ | $\widehat{\beta}_{S=2}$ | $\widehat{\beta}_{S=\|C\|}$ | $\widehat{\beta}^{\theta}_{S=1}$ | $\widehat{\beta}^{\theta}_{S=2}$ | $\widehat{\beta}^{\theta}_{S=\|C\|}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| anneal | 98.75 | 99.60 | 99.52 | 99.57 | 99.51 | 99.25 | 99.45 | 98.79 | 98.71 | 99.45 |
| audiology | 76.69 | 83.41 | 83.67 | 83.28 | 83.33 | 83.59 | 83.44 | 78.32 | 75.90 | 78.37 |
| autos | 76.56 | 78.51 | 78.66 | 79.62 | 77.83 | 78.12 | 79.10 | 70.04 | 69.18 | 74.72 |
| breast-cancer | 75.26 | 72.60 | 73.51 | 73.51 | 72.43 | 72.71 | 72.71 | 72.43 | 72.71 | 72.71 |
| horse-colic | 85.83 | 82.42 | 82.48 | 82.48 | 83.75 | 83.96 | 83.96 | 83.75 | 83.96 | 83.96 |
| german-credit | 71.31 | 70.24 | 70.48 | 70.49 | 69.91 | 70.49 | 70.49 | 69.91 | 70.49 | 70.49 |
| pima-diabetes | 75.10 | 73.58 | 73.89 | 73.89 | 73.58 | 74.31 | 74.31 | 73.58 | 74.31 | 74.31 |
| glass2 | 76.80 | 80.19 | 80.13 | 80.13 | 80.20 | 80.43 | 80.43 | 80.20 | 80.43 | 80.43 |
| hepatitis | 81.18 | 76.33 | 76.40 | 76.40 | 77.62 | 78.87 | 78.87 | 77.62 | 78.87 | 78.87 |
| hypothyroid | 96.85 | 96.61 | 96.61 | 96.70 | 96.54 | 96.61 | 96.75 | 96.52 | 96.66 | 96.65 |
| ionosphere | 89.40 | 87.78 | 88.77 | 88.77 | 88.69 | 89.34 | 89.34 | 88.69 | 89.34 | 89.34 |
| kr-vs-kp | 99.44 | 99.55 | 99.54 | 99.54 | 99.55 | 99.56 | 99.56 | 99.55 | 99.56 | 99.56 |
| labor | 88.63 | 83.17 | 84.57 | 84.57 | 84.10 | 85.43 | 85.43 | 84.10 | 85.43 | 85.43 |
| letter | 80.30 | 82.73 | 82.77 | 84.00 | 82.04 | 82.07 | 83.83 | 74.20 | 73.53 | 80.97 |
| lymphography | 78.08 | 76.11 | 76.05 | 74.24 | 76.92 | 77.53 | 74.97 | 76.31 | 77.07 | 75.91 |
| mfeat | 76.69 | 79.60 | 79.57 | 79.32 | 79.78 | 79.76 | 79.85 | 79.45 | 79.43 | 79.03 |
| mushroom | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
| optdigits | 77.85 | 78.58 | 78.75 | 78.69 | 78.08 | 78.41 | 78.65 | 74.74 | 74.80 | 77.76 |
| segment | 92.32 | 93.29 | 93.39 | 93.40 | 93.27 | 93.29 | 93.28 | 91.96 | 91.81 | 92.81 |
| sick | 93.63 | 92.87 | 92.98 | 92.98 | 93.06 | 93.15 | 93.15 | 93.06 | 93.15 | 93.15 |
| solar-flare | 97.84 | 96.51 | 96.51 | 96.51 | 96.60 | 96.88 | 96.88 | 96.60 | 96.88 | 96.88 |
| sonar | 71.07 | 69.31 | 69.51 | 69.51 | 68.50 | 68.56 | 68.56 | 68.50 | 68.56 | 68.56 |
| soybean | 92.55 | 93.65 | 93.41 | 93.04 | 93.43 | 93.29 | 92.75 | 91.53 | 90.50 | 92.94 |
| sponge | 92.50 | 93.21 | 93.79 | 93.79 | 95.00 | 95.00 | 95.00 | 95.00 | 95.00 | 95.00 |
| vote | 96.27 | 94.48 | 94.96 | 94.96 | 95.52 | 95.68 | 95.68 | 95.52 | 95.68 | 95.68 |
| vowel | 75.11 | 73.85 | 73.76 | 80.67 | 72.80 | 72.76 | 79.99 | 60.79 | 59.95 | 74.23 |
| zoo | 92.61 | 94.95 | 94.95 | 97.82 | 98.11 | 98.11 | 96.72 | 97.72 | 95.54 | 97.91 |
| Average | 85.50 | 85.30 | 85.50 | 85.85 | 85.56 | 85.82 | 86.04 | 84.03 | 83.98 | 85.37 |

Table 6: Detailed Log-likelihood (Full expanded Table 4.2, 4.3 and 4.4 of Chapter 4)

| Dataset | $\star C4.5_\rho$ | $\beta_{S=1}$ | $\beta_{S=2}$ | $\beta_{S=\lvert C\rvert}$ | $\widehat{\beta}_{S=1}$ | $\widehat{\beta}_{S=2}$ | $\widehat{\beta}_{S=\lvert C\rvert}$ | $\widehat{\beta}^{\theta}_{S=1}$ | $\widehat{\beta}^{\theta}_{S=2}$ | $\widehat{\beta}^{\theta}_{S=\lvert C\rvert}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| anneal | -0.23 | -0.22 | -0.07 | -0.06 | -0.21 | -0.11 | -0.09 | -0.22 | -0.22 | -0.17 |
| audiology | -2.08 | -2.00 | -1.21 | -1.27 | -1.95 | -1.23 | -1.24 | -1.91 | -1.90 | -1.52 |
| autos | -1.64 | -1.59 | -1.15 | -1.04 | -1.59 | -1.27 | -1.11 | -1.64 | -1.66 | -1.41 |
| breast-cancer | -0.82 | -0.86 | -0.86 | -0.86 | -0.87 | -0.85 | -0.85 | -0.87 | -0.85 | -0.85 |
| horse-colic | -0.58 | -0.65 | -0.64 | -0.64 | -0.62 | -0.60 | -0.60 | -0.62 | -0.60 | -0.60 |
| german-credit | -0.84 | -0.86 | -0.88 | -0.88 | -0.87 | -0.83 | -0.83 | -0.87 | -0.83 | -0.83 |
| pima-diabetes | -0.78 | -0.81 | -0.81 | -0.81 | -0.80 | -0.78 | -0.78 | -0.80 | -0.78 | -0.78 |
| glass2 | -0.74 | -0.67 | -0.67 | -0.67 | -0.67 | -0.67 | -0.67 | -0.67 | -0.67 | -0.67 |
| hepatitis | -0.67 | -0.73 | -0.75 | -0.75 | -0.70 | -0.67 | -0.67 | -0.70 | -0.67 | -0.67 |
| hypothyroid | -0.13 | -0.14 | -0.13 | -0.13 | -0.14 | -0.13 | -0.13 | -0.14 | -0.13 | -0.13 |
| ionosphere | -0.44 | -0.44 | -0.45 | -0.45 | -0.44 | -0.43 | -0.43 | -0.44 | -0.43 | -0.43 |
| kr-vs-kp | -0.05 | -0.04 | -0.03 | -0.03 | -0.04 | -0.04 | -0.04 | -0.04 | -0.04 | -0.04 |
| labor | -0.45 | -0.58 | -0.58 | -0.58 | -0.55 | -0.54 | -0.54 | -0.55 | -0.54 | -0.54 |
| letter | -1.86 | -1.77 | -1.07 | -0.99 | -1.74 | -1.10 | -0.99 | -1.71 | -1.71 | -1.38 |
| lymphography | -0.90 | -0.96 | -0.94 | -0.96 | -0.93 | -0.89 | -0.89 | -0.93 | -0.90 | -0.86 |
| mfeat | -1.30 | -1.21 | -1.18 | -1.16 | -1.20 | -1.14 | -1.09 | -1.20 | -1.23 | -1.14 |
| mushroom | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 | -0.00 |
| optdigits | -1.35 | -1.40 | -1.19 | -1.12 | -1.38 | -1.17 | -1.08 | -1.39 | -1.39 | -1.21 |
| segment | -0.51 | -0.47 | -0.34 | -0.32 | -0.47 | -0.36 | -0.33 | -0.48 | -0.49 | -0.42 |
| sick | -0.31 | -0.21 | -0.21 | -0.21 | -0.21 | -0.21 | -0.21 | -0.21 | -0.21 | -0.21 |
| solar-flare | -0.15 | -0.18 | -0.18 | -0.18 | -0.17 | -0.17 | -0.17 | -0.17 | -0.17 | -0.17 |
| sonar | -0.89 | -0.92 | -0.97 | -0.97 | -0.94 | -0.91 | -0.91 | -0.94 | -0.91 | -0.91 |
| soybean | -1.16 | -1.23 | -0.41 | -0.36 | -1.22 | -0.47 | -0.42 | -1.15 | -1.16 | -0.83 |
| sponge | -0.48 | -0.43 | -0.43 | -0.43 | -0.38 | -0.38 | -0.38 | -0.38 | -0.38 | -0.38 |
| vote | -0.20 | -0.25 | -0.25 | -0.25 | -0.23 | -0.22 | -0.22 | -0.23 | -0.22 | -0.22 |
| vowel | -2.07 | -1.93 | -1.30 | -1.07 | -1.92 | -1.41 | -1.17 | -1.94 | -1.95 | -1.62 |
| zoo | -0.68 | -0.80 | -0.38 | -0.25 | -0.74 | -0.39 | -0.37 | -0.75 | -0.77 | -0.60 |
| Average | -0.79 | -0.79 | -0.63 | -0.61 | -0.78 | -0.63 | -0.60 | -0.78 | -0.77 | -0.69 |

Table 7: Detailed Error for BRS ensembles M=1, 3; and Trees=10, 50, 100, 200; ($BRS^{M}_{Trees}$) - Section 4.4.4

| Dataset | $BRS^{1}_{10}$ | $BRS^{1}_{50}$ | $BRS^{1}_{100}$ | $BRS^{1}_{200}$ | $BRS^{3}_{10}$ | $BRS^{3}_{50}$ | $BRS^{3}_{100}$ | $BRS^{3}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.010 | 0.009 | 0.009 | 0.009 | 0.010 | 0.008 | 0.008 | 0.008 |
| audiology.arff | 0.269 | 0.250 | 0.239 | 0.238 | 0.263 | 0.238 | 0.239 | 0.238 |
| autos.arff | 0.306 | 0.273 | 0.264 | 0.264 | 0.278 | 0.262 | 0.259 | 0.256 |
| breast-cancer.arff | 0.280 | 0.276 | 0.273 | 0.277 | 0.277 | 0.275 | 0.274 | 0.275 |
| colic.arff | 0.199 | 0.187 | 0.182 | 0.183 | 0.195 | 0.179 | 0.181 | 0.178 |
| credit-g.arff | 0.278 | 0.262 | 0.262 | 0.262 | 0.276 | 0.264 | 0.265 | 0.261 |
| diabetes.arff | 0.276 | 0.274 | 0.272 | 0.269 | 0.268 | 0.265 | 0.267 | 0.266 |
| glass2.arff | 0.208 | 0.184 | 0.190 | 0.185 | 0.197 | 0.188 | 0.180 | 0.185 |
| hepatitis.arff | 0.185 | 0.181 | 0.180 | 0.181 | 0.194 | 0.190 | 0.187 | 0.188 |
| hypothyroid.arff | 0.043 | 0.042 | 0.042 | 0.042 | 0.042 | 0.041 | 0.041 | 0.041 |
| ionosphere.arff | 0.099 | 0.091 | 0.089 | 0.087 | 0.094 | 0.087 | 0.086 | 0.084 |
| kr-vs-kp.arff | 0.020 | 0.015 | 0.014 | 0.014 | 0.014 | 0.010 | 0.010 | 0.009 |
| labor.arff | 0.118 | 0.104 | 0.102 | 0.091 | 0.132 | 0.107 | 0.104 | 0.107 |
| lymph.arff | 0.193 | 0.172 | 0.166 | 0.164 | 0.184 | 0.170 | 0.176 | 0.175 |
| segment.arff | 0.063 | 0.052 | 0.052 | 0.051 | 0.060 | 0.054 | 0.054 | 0.052 |
| sick.arff | 0.065 | 0.064 | 0.064 | 0.064 | 0.064 | 0.064 | 0.064 | 0.064 |
| solar-flare-1.arff | 0.029 | 0.029 | 0.029 | 0.029 | 0.032 | 0.030 | 0.030 | 0.030 |
| sonar.arff | 0.282 | 0.238 | 0.234 | 0.233 | 0.257 | 0.233 | 0.233 | 0.227 |
| soybean.arff | 0.070 | 0.063 | 0.061 | 0.061 | 0.067 | 0.061 | 0.062 | 0.063 |
| sponge.arff | 0.068 | 0.064 | 0.063 | 0.063 | 0.066 | 0.063 | 0.064 | 0.063 |
| vote.arff | 0.050 | 0.044 | 0.041 | 0.041 | 0.046 | 0.042 | 0.041 | 0.040 |
| vowel.arff | 0.220 | 0.181 | 0.172 | 0.169 | 0.238 | 0.206 | 0.202 | 0.199 |
| zoo.arff | 0.058 | 0.057 | 0.056 | 0.057 | 0.076 | 0.067 | 0.066 | 0.065 |
| Average | 0.147 | 0.135 | 0.133 | 0.132 | 0.145 | 0.135 | 0.134 | 0.134 |

Table 8: Detailed for Error BRS ensembles M=5, Log N; and Trees=10, 50, 100, 200; ($BRS^{M}_{Trees}$) - Section 4.4.4

| Dataset | $BRS^{5}_{10}$ | $BRS^{5}_{50}$ | $BRS^{5}_{100}$ | $BRS^{5}_{200}$ | $BRS^{LogN}_{10}$ | $BRS^{LogN}_{50}$ | $BRS^{LogN}_{100}$ | $BRS^{LogN}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.010 | 0.008 | 0.008 | 0.008 | 0.010 | 0.008 | 0.008 | 0.008 |
| audiology.arff | 0.250 | 0.225 | 0.229 | 0.233 | 0.248 | 0.233 | 0.227 | 0.227 |
| autos.arff | 0.280 | 0.255 | 0.254 | 0.251 | 0.280 | 0.255 | 0.254 | 0.251 |
| breast-cancer.arff | 0.284 | 0.273 | 0.278 | 0.278 | 0.282 | 0.279 | 0.274 | 0.272 |
| colic.arff | 0.194 | 0.174 | 0.178 | 0.177 | 0.194 | 0.174 | 0.178 | 0.177 |
| credit-g.arff | 0.276 | 0.266 | 0.265 | 0.264 | 0.276 | 0.266 | 0.265 | 0.264 |
| diabetes.arff | 0.267 | 0.266 | 0.264 | 0.264 | 0.267 | 0.264 | 0.264 | 0.264 |
| glass2.arff | 0.210 | 0.195 | 0.190 | 0.196 | 0.208 | 0.191 | 0.183 | 0.186 |
| hepatitis.arff | 0.201 | 0.190 | 0.190 | 0.185 | 0.201 | 0.190 | 0.190 | 0.185 |
| hypothyroid.arff | 0.041 | 0.040 | 0.040 | 0.040 | 0.041 | 0.040 | 0.040 | 0.040 |
| ionosphere.arff | 0.092 | 0.084 | 0.083 | 0.082 | 0.089 | 0.084 | 0.084 | 0.085 |
| kr-vs-kp.arff | 0.011 | 0.009 | 0.009 | 0.009 | 0.010 | 0.008 | 0.009 | 0.008 |
| labor.arff | 0.137 | 0.125 | 0.126 | 0.132 | 0.137 | 0.125 | 0.126 | 0.132 |
| lymph.arff | 0.195 | 0.189 | 0.186 | 0.177 | 0.195 | 0.189 | 0.186 | 0.177 |
| segment.arff | 0.060 | 0.055 | 0.055 | 0.055 | 0.060 | 0.055 | 0.055 | 0.055 |
| sick.arff | 0.065 | 0.064 | 0.065 | 0.064 | 0.065 | 0.064 | 0.065 | 0.064 |
| solar-flare-1.arff | 0.029 | 0.029 | 0.029 | 0.029 | 0.030 | 0.029 | 0.029 | 0.028 |
| sonar.arff | 0.276 | 0.237 | 0.237 | 0.241 | 0.288 | 0.238 | 0.241 | 0.242 |
| soybean.arff | 0.068 | 0.066 | 0.065 | 0.065 | 0.072 | 0.066 | 0.065 | 0.065 |
| sponge.arff | 0.064 | 0.063 | 0.063 | 0.063 | 0.070 | 0.067 | 0.064 | 0.064 |
| vote.arff | 0.043 | 0.040 | 0.041 | 0.040 | 0.043 | 0.040 | 0.041 | 0.040 |
| vowel.arff | 0.258 | 0.235 | 0.233 | 0.235 | 0.249 | 0.218 | 0.217 | 0.216 |
| zoo.arff | 0.078 | 0.074 | 0.075 | 0.076 | 0.078 | 0.074 | 0.075 | 0.076 |
| Average | 0.147 | 0.138 | 0.137 | 0.138 | 0.148 | 0.137 | 0.137 | 0.136 |

Table 9: Detailed Bias for BRS ensembles with M=1, 3; and Trees=10, 50, 100, 200; ($BRS^{M}_{Trees}$) - Section 4.4.4

| Dataset | $BRS^{1}_{10}$ | $BRS^{1}_{50}$ | $BRS^{1}_{100}$ | $BRS^{1}_{200}$ | $BRS^{3}_{10}$ | $BRS^{3}_{50}$ | $BRS^{3}_{100}$ | $BRS^{3}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.003 | 0.004 | 0.004 | 0.004 | 0.004 | 0.003 | 0.004 | 0.003 |
| audiology.arff | 0.102 | 0.127 | 0.127 | 0.127 | 0.108 | 0.114 | 0.122 | 0.123 |
| autos.arff | 0.106 | 0.116 | 0.120 | 0.120 | 0.105 | 0.108 | 0.111 | 0.113 |
| breast-cancer.arff | 0.213 | 0.216 | 0.214 | 0.218 | 0.215 | 0.215 | 0.211 | 0.217 |
| colic.arff | 0.139 | 0.153 | 0.158 | 0.160 | 0.137 | 0.144 | 0.147 | 0.147 |
| credit-g.arff | 0.208 | 0.212 | 0.215 | 0.219 | 0.199 | 0.209 | 0.214 | 0.213 |
| diabetes.arff | 0.201 | 0.211 | 0.211 | 0.208 | 0.198 | 0.200 | 0.207 | 0.206 |
| glass2.arff | 0.099 | 0.086 | 0.092 | 0.095 | 0.083 | 0.091 | 0.088 | 0.098 |
| hepatitis.arff | 0.141 | 0.146 | 0.146 | 0.145 | 0.145 | 0.149 | 0.145 | 0.153 |
| hypothyroid.arff | 0.026 | 0.027 | 0.027 | 0.027 | 0.024 | 0.026 | 0.025 | 0.025 |
| ionosphere.arff | 0.072 | 0.072 | 0.075 | 0.073 | 0.064 | 0.068 | 0.066 | 0.065 |
| kr-vs-kp.arff | 0.007 | 0.007 | 0.007 | 0.007 | 0.005 | 0.005 | 0.005 | 0.005 |
| labor.arff | 0.033 | 0.038 | 0.036 | 0.030 | 0.029 | 0.046 | 0.034 | 0.040 |
| lymph.arff | 0.110 | 0.114 | 0.112 | 0.117 | 0.109 | 0.107 | 0.112 | 0.115 |
| segment.arff | 0.030 | 0.029 | 0.029 | 0.030 | 0.028 | 0.030 | 0.031 | 0.031 |
| sick.arff | 0.051 | 0.053 | 0.053 | 0.053 | 0.052 | 0.052 | 0.053 | 0.053 |
| solar-flare-1.arff | 0.026 | 0.026 | 0.026 | 0.025 | 0.026 | 0.025 | 0.026 | 0.026 |
| sonar.arff | 0.154 | 0.163 | 0.161 | 0.169 | 0.140 | 0.154 | 0.169 | 0.166 |
| soybean.arff | 0.040 | 0.043 | 0.044 | 0.043 | 0.038 | 0.040 | 0.042 | 0.043 |
| sponge.arff | 0.053 | 0.053 | 0.053 | 0.053 | 0.053 | 0.053 | 0.053 | 0.053 |
| vote.arff | 0.031 | 0.031 | 0.031 | 0.030 | 0.030 | 0.031 | 0.032 | 0.032 |
| vowel.arff | 0.052 | 0.050 | 0.050 | 0.052 | 0.060 | 0.066 | 0.068 | 0.068 |
| zoo.arff | 0.017 | 0.020 | 0.026 | 0.025 | 0.029 | 0.025 | 0.026 | 0.024 |
| Average | 0.083 | 0.087 | 0.088 | 0.088 | 0.082 | 0.085 | 0.087 | 0.088 |

Table 10: Detailed Bias for BRS ensembles M=5, Log N; and Trees=10, 50, 100, 200; ($BRS^{M}_{Trees}$) - Section 4.4.4

| Dataset | $BRS^{5}_{10}$ | $BRS^{5}_{50}$ | $BRS^{5}_{100}$ | $BRS^{5}_{200}$ | $BRS^{LogN}_{10}$ | $BRS^{LogN}_{50}$ | $BRS^{LogN}_{100}$ | $BRS^{LogN}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.004 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 | 0.003 |
| audiology.arff | 0.102 | 0.109 | 0.109 | 0.114 | 0.106 | 0.114 | 0.108 | 0.110 |
| autos.arff | 0.099 | 0.107 | 0.107 | 0.108 | 0.099 | 0.107 | 0.107 | 0.108 |
| breast-cancer.arff | 0.214 | 0.213 | 0.214 | 0.215 | 0.220 | 0.222 | 0.217 | 0.212 |
| colic.arff | 0.146 | 0.140 | 0.147 | 0.147 | 0.146 | 0.140 | 0.147 | 0.147 |
| credit-g.arff | 0.200 | 0.210 | 0.213 | 0.213 | 0.200 | 0.210 | 0.213 | 0.213 |
| diabetes.arff | 0.202 | 0.204 | 0.205 | 0.205 | 0.191 | 0.198 | 0.205 | 0.206 |
| glass2.arff | 0.083 | 0.082 | 0.076 | 0.087 | 0.094 | 0.096 | 0.089 | 0.089 |
| hepatitis.arff | 0.157 | 0.151 | 0.155 | 0.149 | 0.157 | 0.151 | 0.155 | 0.149 |
| hypothyroid.arff | 0.024 | 0.024 | 0.024 | 0.024 | 0.024 | 0.024 | 0.024 | 0.024 |
| ionosphere.arff | 0.065 | 0.063 | 0.063 | 0.062 | 0.063 | 0.063 | 0.061 | 0.062 |
| kr-vs-kp.arff | 0.005 | 0.005 | 0.006 | 0.006 | 0.005 | 0.005 | 0.005 | 0.005 |
| labor.arff | 0.053 | 0.045 | 0.049 | 0.055 | 0.053 | 0.045 | 0.049 | 0.055 |
| lymph.arff | 0.120 | 0.120 | 0.115 | 0.120 | 0.120 | 0.120 | 0.115 | 0.120 |
| segment.arff | 0.030 | 0.032 | 0.032 | 0.032 | 0.030 | 0.032 | 0.032 | 0.032 |
| sick.arff | 0.053 | 0.052 | 0.053 | 0.053 | 0.053 | 0.052 | 0.053 | 0.053 |
| solar-flare-1.arff | 0.024 | 0.025 | 0.025 | 0.025 | 0.026 | 0.025 | 0.026 | 0.025 |
| sonar.arff | 0.149 | 0.160 | 0.177 | 0.179 | 0.171 | 0.168 | 0.173 | 0.177 |
| soybean.arff | 0.040 | 0.040 | 0.040 | 0.041 | 0.043 | 0.040 | 0.039 | 0.040 |
| sponge.arff | 0.051 | 0.053 | 0.053 | 0.053 | 0.053 | 0.053 | 0.053 | 0.053 |
| vote.arff | 0.033 | 0.033 | 0.032 | 0.032 | 0.033 | 0.033 | 0.032 | 0.032 |
| vowel.arff | 0.079 | 0.083 | 0.083 | 0.084 | 0.075 | 0.074 | 0.079 | 0.078 |
| zoo.arff | 0.030 | 0.032 | 0.029 | 0.028 | 0.030 | 0.032 | 0.029 | 0.028 |
| Average | 0.085 | 0.086 | 0.087 | 0.089 | 0.087 | 0.087 | 0.088 | 0.088 |

Table 11: Detailed Variance for BRS ensembles with M=1, 3; and Trees=10, 50, 100, 200; ($BRS^{M}_{Trees}$) - Section 4.4.4

| Dataset | $BRS^{1}_{10}$ | $BRS^{1}_{50}$ | $BRS^{1}_{100}$ | $BRS^{1}_{200}$ | $BRS^{3}_{10}$ | $BRS^{3}_{50}$ | $BRS^{3}_{100}$ | $BRS^{3}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.007 | 0.005 | 0.005 | 0.005 | 0.006 | 0.005 | 0.005 | 0.005 |
| audiology.arff | 0.167 | 0.123 | 0.112 | 0.111 | 0.155 | 0.124 | 0.117 | 0.114 |
| autos.arff | 0.200 | 0.158 | 0.145 | 0.145 | 0.173 | 0.154 | 0.148 | 0.143 |
| breast-cancer.arff | 0.067 | 0.059 | 0.059 | 0.059 | 0.062 | 0.059 | 0.063 | 0.058 |
| colic.arff | 0.060 | 0.033 | 0.025 | 0.024 | 0.059 | 0.035 | 0.034 | 0.031 |
| credit-g.arff | 0.070 | 0.050 | 0.047 | 0.043 | 0.077 | 0.055 | 0.051 | 0.048 |
| diabetes.arff | 0.075 | 0.063 | 0.061 | 0.062 | 0.070 | 0.065 | 0.060 | 0.060 |
| glass2.arff | 0.109 | 0.098 | 0.097 | 0.090 | 0.114 | 0.097 | 0.092 | 0.088 |
| hepatitis.arff | 0.044 | 0.035 | 0.034 | 0.036 | 0.049 | 0.041 | 0.042 | 0.035 |
| hypothyroid.arff | 0.017 | 0.015 | 0.015 | 0.015 | 0.017 | 0.015 | 0.015 | 0.015 |
| ionosphere.arff | 0.027 | 0.019 | 0.014 | 0.014 | 0.030 | 0.019 | 0.020 | 0.020 |
| kr-vs-kp.arff | 0.013 | 0.007 | 0.007 | 0.007 | 0.009 | 0.005 | 0.005 | 0.004 |
| labor.arff | 0.084 | 0.066 | 0.066 | 0.061 | 0.103 | 0.061 | 0.069 | 0.067 |
| lymph.arff | 0.082 | 0.059 | 0.053 | 0.048 | 0.075 | 0.062 | 0.065 | 0.060 |
| segment.arff | 0.032 | 0.023 | 0.022 | 0.021 | 0.032 | 0.024 | 0.023 | 0.021 |
| sick.arff | 0.013 | 0.011 | 0.011 | 0.011 | 0.012 | 0.011 | 0.011 | 0.011 |
| solar-flare-1.arff | 0.004 | 0.003 | 0.003 | 0.004 | 0.006 | 0.005 | 0.004 | 0.004 |
| sonar.arff | 0.128 | 0.076 | 0.074 | 0.065 | 0.118 | 0.079 | 0.065 | 0.061 |
| soybean.arff | 0.030 | 0.020 | 0.017 | 0.018 | 0.029 | 0.021 | 0.020 | 0.020 |
| sponge.arff | 0.016 | 0.012 | 0.011 | 0.011 | 0.013 | 0.011 | 0.012 | 0.011 |
| vote.arff | 0.020 | 0.013 | 0.010 | 0.011 | 0.016 | 0.011 | 0.009 | 0.008 |
| vowel.arff | 0.168 | 0.131 | 0.122 | 0.117 | 0.178 | 0.140 | 0.133 | 0.131 |
| zoo.arff | 0.042 | 0.038 | 0.031 | 0.032 | 0.048 | 0.043 | 0.041 | 0.042 |
| Average | 0.064 | 0.049 | 0.045 | 0.044 | 0.063 | 0.050 | 0.048 | 0.046 |

Table 12: Detailed Variance for BRS ensembles M=5, Log N; and Trees=10, 50, 100, 200; ($BRS^{M}_{Trees}$) - Section 4.4.4

| Dataset | $BRS^{5}_{10}$ | $BRS^{5}_{50}$ | $BRS^{5}_{100}$ | $BRS^{5}_{200}$ | $BRS^{LogN}_{10}$ | $BRS^{LogN}_{50}$ | $BRS^{LogN}_{100}$ | $BRS^{LogN}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.006 | 0.005 | 0.005 | 0.005 | 0.007 | 0.005 | 0.005 | 0.005 |
| audiology.arff | 0.148 | 0.116 | 0.120 | 0.119 | 0.142 | 0.119 | 0.119 | 0.117 |
| autos.arff | 0.181 | 0.148 | 0.147 | 0.143 | 0.181 | 0.148 | 0.147 | 0.143 |
| breast-cancer.arff | 0.069 | 0.060 | 0.064 | 0.063 | 0.062 | 0.057 | 0.057 | 0.060 |
| colic.arff | 0.048 | 0.035 | 0.031 | 0.030 | 0.048 | 0.035 | 0.031 | 0.030 |
| credit-g.arff | 0.076 | 0.056 | 0.052 | 0.051 | 0.076 | 0.056 | 0.052 | 0.051 |
| diabetes.arff | 0.065 | 0.062 | 0.059 | 0.059 | 0.076 | 0.065 | 0.059 | 0.058 |
| glass2.arff | 0.128 | 0.113 | 0.114 | 0.109 | 0.114 | 0.096 | 0.094 | 0.097 |
| hepatitis.arff | 0.044 | 0.039 | 0.035 | 0.036 | 0.044 | 0.039 | 0.035 | 0.036 |
| hypothyroid.arff | 0.017 | 0.016 | 0.016 | 0.015 | 0.017 | 0.016 | 0.016 | 0.015 |
| ionosphere.arff | 0.027 | 0.021 | 0.020 | 0.021 | 0.027 | 0.021 | 0.022 | 0.023 |
| kr-vs-kp.arff | 0.006 | 0.004 | 0.003 | 0.003 | 0.005 | 0.004 | 0.004 | 0.003 |
| labor.arff | 0.084 | 0.080 | 0.077 | 0.076 | 0.084 | 0.080 | 0.077 | 0.076 |
| lymph.arff | 0.076 | 0.069 | 0.071 | 0.057 | 0.076 | 0.069 | 0.071 | 0.057 |
| segment.arff | 0.030 | 0.023 | 0.023 | 0.023 | 0.030 | 0.023 | 0.023 | 0.023 |
| sick.arff | 0.012 | 0.012 | 0.011 | 0.011 | 0.012 | 0.012 | 0.011 | 0.011 |
| solar-flare-1.arff | 0.005 | 0.004 | 0.004 | 0.005 | 0.004 | 0.004 | 0.003 | 0.003 |
| sonar.arff | 0.127 | 0.077 | 0.060 | 0.062 | 0.117 | 0.069 | 0.068 | 0.065 |
| soybean.arff | 0.028 | 0.026 | 0.025 | 0.024 | 0.029 | 0.025 | 0.026 | 0.025 |
| sponge.arff | 0.013 | 0.011 | 0.011 | 0.011 | 0.017 | 0.014 | 0.012 | 0.012 |
| vote.arff | 0.011 | 0.008 | 0.009 | 0.008 | 0.011 | 0.008 | 0.009 | 0.008 |
| vowel.arff | 0.179 | 0.152 | 0.150 | 0.150 | 0.174 | 0.145 | 0.139 | 0.139 |
| zoo.arff | 0.049 | 0.043 | 0.046 | 0.048 | 0.049 | 0.043 | 0.046 | 0.048 |
| Average | 0.062 | 0.051 | 0.050 | 0.049 | 0.061 | 0.050 | 0.049 | 0.048 |

Table 13: Detailed Error for RF ensembles M=1, 3; and Trees=10, 50, 100, 200; ($RF^{M}_{Trees}$) - Section 4.4.4

| Dataset | $RF^{1}_{10}$ | $RF^{1}_{50}$ | $RF^{1}_{100}$ | $RF^{1}_{200}$ | $RF^{3}_{10}$ | $RF^{3}_{50}$ | $RF^{3}_{100}$ | $RF^{3}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.034 | 0.021 | 0.020 | 0.019 | 0.025 | 0.019 | 0.016 | 0.016 |
| audiology.arff | 0.348 | 0.305 | 0.300 | 0.298 | 0.356 | 0.307 | 0.301 | 0.300 |
| autos.arff | 0.315 | 0.308 | 0.298 | 0.298 | 0.293 | 0.269 | 0.272 | 0.272 |
| breast-cancer.arff | 0.297 | 0.290 | 0.292 | 0.291 | 0.324 | 0.320 | 0.312 | 0.312 |
| colic.arff | 0.248 | 0.202 | 0.198 | 0.198 | 0.208 | 0.187 | 0.182 | 0.180 |
| credit-g.arff | 0.290 | 0.280 | 0.278 | 0.277 | 0.280 | 0.269 | 0.269 | 0.267 |
| diabetes.arff | 0.294 | 0.275 | 0.276 | 0.273 | 0.281 | 0.269 | 0.267 | 0.269 |
| glass2.arff | 0.243 | 0.206 | 0.202 | 0.196 | 0.204 | 0.193 | 0.182 | 0.185 |
| hepatitis.arff | 0.181 | 0.179 | 0.174 | 0.181 | 0.189 | 0.175 | 0.172 | 0.177 |
| hypothyroid.arff | 0.060 | 0.058 | 0.058 | 0.058 | 0.057 | 0.055 | 0.055 | 0.055 |
| ionosphere.arff | 0.108 | 0.100 | 0.098 | 0.096 | 0.104 | 0.089 | 0.089 | 0.089 |
| kr-vs-kp.arff | 0.067 | 0.035 | 0.030 | 0.028 | 0.032 | 0.019 | 0.017 | 0.017 |
| labor.arff | 0.140 | 0.089 | 0.082 | 0.082 | 0.156 | 0.116 | 0.111 | 0.100 |
| lymph.arff | 0.214 | 0.169 | 0.170 | 0.161 | 0.213 | 0.174 | 0.172 | 0.167 |
| segment.arff | 0.081 | 0.064 | 0.062 | 0.060 | 0.069 | 0.054 | 0.052 | 0.051 |
| sick.arff | 0.064 | 0.063 | 0.063 | 0.063 | 0.064 | 0.063 | 0.064 | 0.064 |
| solar-flare-1.arff | 0.027 | 0.029 | 0.027 | 0.027 | 0.029 | 0.030 | 0.029 | 0.029 |
| sonar.arff | 0.335 | 0.267 | 0.249 | 0.235 | 0.310 | 0.239 | 0.234 | 0.232 |
| soybean.arff | 0.116 | 0.086 | 0.081 | 0.080 | 0.104 | 0.080 | 0.076 | 0.075 |
| sponge.arff | 0.064 | 0.063 | 0.063 | 0.063 | 0.063 | 0.063 | 0.063 | 0.063 |
| vote.arff | 0.064 | 0.049 | 0.047 | 0.048 | 0.047 | 0.040 | 0.040 | 0.039 |
| vowel.arff | 0.244 | 0.167 | 0.156 | 0.150 | 0.223 | 0.165 | 0.160 | 0.156 |
| zoo.arff | 0.053 | 0.061 | 0.055 | 0.055 | 0.069 | 0.062 | 0.054 | 0.058 |
| Average | 0.169 | 0.146 | 0.143 | 0.141 | 0.161 | 0.142 | 0.139 | 0.138 |

Table 14: Detailed Error for RF ensembles M=5, Log N; and Trees=10, 50, 100, 200; ($RF^{M}_{Trees}$) - Section 4.4.4

| Dataset | $RF^{5}_{10}$ | $RF^{5}_{50}$ | $RF^{5}_{100}$ | $RF^{5}_{200}$ | $RF^{LogN}_{10}$ | $RF^{LogN}_{50}$ | $RF^{LogN}_{100}$ | $RF^{LogN}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.021 | 0.014 | 0.013 | 0.012 | 0.018 | 0.014 | 0.012 | 0.012 |
| audiology.arff | 0.346 | 0.302 | 0.298 | 0.300 | 0.351 | 0.295 | 0.295 | 0.297 |
| autos.arff | 0.287 | 0.262 | 0.266 | 0.260 | 0.287 | 0.262 | 0.266 | 0.260 |
| breast-cancer.arff | 0.332 | 0.326 | 0.327 | 0.320 | 0.330 | 0.321 | 0.316 | 0.317 |
| colic.arff | 0.190 | 0.169 | 0.168 | 0.171 | 0.190 | 0.169 | 0.168 | 0.171 |
| credit-g.arff | 0.284 | 0.268 | 0.266 | 0.264 | 0.284 | 0.268 | 0.266 | 0.264 |
| diabetes.arff | 0.285 | 0.272 | 0.269 | 0.269 | 0.277 | 0.267 | 0.268 | 0.266 |
| glass2.arff | 0.210 | 0.191 | 0.186 | 0.187 | 0.212 | 0.190 | 0.187 | 0.182 |
| hepatitis.arff | 0.206 | 0.188 | 0.181 | 0.183 | 0.206 | 0.188 | 0.181 | 0.183 |
| hypothyroid.arff | 0.055 | 0.053 | 0.052 | 0.052 | 0.055 | 0.053 | 0.052 | 0.052 |
| ionosphere.arff | 0.103 | 0.089 | 0.086 | 0.086 | 0.105 | 0.089 | 0.089 | 0.088 |
| kr-vs-kp.arff | 0.021 | 0.014 | 0.013 | 0.013 | 0.019 | 0.013 | 0.013 | 0.012 |
| labor.arff | 0.146 | 0.133 | 0.126 | 0.126 | 0.146 | 0.133 | 0.126 | 0.126 |
| lymph.arff | 0.205 | 0.183 | 0.176 | 0.174 | 0.205 | 0.183 | 0.176 | 0.174 |
| segment.arff | 0.062 | 0.050 | 0.050 | 0.049 | 0.062 | 0.050 | 0.050 | 0.049 |
| sick.arff | 0.065 | 0.064 | 0.064 | 0.064 | 0.065 | 0.064 | 0.064 | 0.064 |
| solar-flare-1.arff | 0.032 | 0.031 | 0.032 | 0.032 | 0.032 | 0.032 | 0.030 | 0.031 |
| sonar.arff | 0.293 | 0.251 | 0.235 | 0.229 | 0.296 | 0.250 | 0.244 | 0.233 |
| soybean.arff | 0.100 | 0.081 | 0.078 | 0.079 | 0.105 | 0.081 | 0.082 | 0.081 |
| sponge.arff | 0.062 | 0.063 | 0.063 | 0.063 | 0.066 | 0.063 | 0.063 | 0.063 |
| vote.arff | 0.049 | 0.040 | 0.040 | 0.039 | 0.049 | 0.040 | 0.040 | 0.039 |
| vowel.arff | 0.223 | 0.189 | 0.183 | 0.182 | 0.222 | 0.174 | 0.167 | 0.169 |
| zoo.arff | 0.082 | 0.078 | 0.073 | 0.073 | 0.082 | 0.078 | 0.073 | 0.073 |
| Average | 0.159 | 0.144 | 0.141 | 0.140 | 0.159 | 0.143 | 0.140 | 0.139 |

Table 15: Detailed Bias for RF ensembles M=1, 3; and Trees=10, 50, 100, 200; ($RF^{M}_{Trees}$) - Section 4.4.4

| Dataset | $RF^{1}_{10}$ | $RF^{1}_{50}$ | $RF^{1}_{100}$ | $RF^{1}_{200}$ | $RF^{3}_{10}$ | $RF^{3}_{50}$ | $RF^{3}_{100}$ | $RF^{3}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.009 | 0.008 | 0.009 | 0.008 | 0.006 | 0.006 | 0.005 | 0.005 |
| audiology.arff | 0.124 | 0.153 | 0.158 | 0.160 | 0.123 | 0.152 | 0.162 | 0.158 |
| autos.arff | 0.142 | 0.174 | 0.168 | 0.171 | 0.122 | 0.134 | 0.133 | 0.137 |
| breast-cancer.arff | 0.224 | 0.228 | 0.230 | 0.233 | 0.237 | 0.239 | 0.230 | 0.228 |
| colic.arff | 0.155 | 0.155 | 0.158 | 0.163 | 0.128 | 0.147 | 0.146 | 0.149 |
| credit-g.arff | 0.224 | 0.250 | 0.252 | 0.252 | 0.209 | 0.227 | 0.231 | 0.230 |
| diabetes.arff | 0.206 | 0.207 | 0.213 | 0.212 | 0.197 | 0.202 | 0.199 | 0.207 |
| glass2.arff | 0.117 | 0.125 | 0.114 | 0.115 | 0.082 | 0.095 | 0.095 | 0.102 |
| hepatitis.arff | 0.134 | 0.152 | 0.144 | 0.147 | 0.137 | 0.136 | 0.136 | 0.141 |
| hypothyroid.arff | 0.046 | 0.048 | 0.048 | 0.048 | 0.042 | 0.045 | 0.044 | 0.045 |
| ionosphere.arff | 0.077 | 0.084 | 0.086 | 0.086 | 0.070 | 0.072 | 0.075 | 0.075 |
| kr-vs-kp.arff | 0.020 | 0.015 | 0.013 | 0.013 | 0.011 | 0.009 | 0.008 | 0.009 |
| labor.arff | 0.022 | 0.030 | 0.015 | 0.015 | 0.068 | 0.050 | 0.048 | 0.048 |
| lymph.arff | 0.118 | 0.131 | 0.130 | 0.128 | 0.141 | 0.132 | 0.133 | 0.131 |
| segment.arff | 0.033 | 0.036 | 0.037 | 0.036 | 0.028 | 0.029 | 0.029 | 0.030 |
| sick.arff | 0.054 | 0.055 | 0.055 | 0.055 | 0.052 | 0.053 | 0.054 | 0.054 |
| solar-flare-1.arff | 0.022 | 0.023 | 0.023 | 0.023 | 0.022 | 0.025 | 0.025 | 0.024 |
| sonar.arff | 0.161 | 0.157 | 0.166 | 0.159 | 0.160 | 0.152 | 0.161 | 0.161 |
| soybean.arff | 0.051 | 0.056 | 0.056 | 0.058 | 0.051 | 0.053 | 0.053 | 0.054 |
| sponge.arff | 0.053 | 0.053 | 0.053 | 0.053 | 0.051 | 0.053 | 0.053 | 0.053 |
| vote.arff | 0.037 | 0.032 | 0.035 | 0.038 | 0.029 | 0.032 | 0.033 | 0.033 |
| vowel.arff | 0.036 | 0.039 | 0.039 | 0.036 | 0.046 | 0.042 | 0.043 | 0.043 |
| zoo.arff | 0.019 | 0.023 | 0.024 | 0.024 | 0.021 | 0.017 | 0.022 | 0.017 |
| Average | 0.091 | 0.097 | 0.097 | 0.097 | 0.088 | 0.091 | 0.092 | 0.093 |

Table 16: Detailed Bias for RF ensembles M=5, Log N; and Trees=10, 50, 100, 200; ($RF^{M}_{Trees}$) - Section 4.4.4

| Dataset | $RF^{5}_{10}$ | $RF^{5}_{50}$ | $RF^{5}_{100}$ | $RF^{5}_{200}$ | $RF^{LogN}_{10}$ | $RF^{LogN}_{50}$ | $RF^{LogN}_{100}$ | $RF^{LogN}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.005 | 0.005 | 0.005 | 0.004 | 0.005 | 0.004 | 0.004 | 0.004 |
| audiology.arff | 0.135 | 0.153 | 0.154 | 0.161 | 0.150 | 0.145 | 0.156 | 0.158 |
| autos.arff | 0.115 | 0.113 | 0.120 | 0.112 | 0.115 | 0.113 | 0.120 | 0.112 |
| breast-cancer.arff | 0.232 | 0.226 | 0.234 | 0.227 | 0.237 | 0.235 | 0.230 | 0.227 |
| colic.arff | 0.135 | 0.135 | 0.138 | 0.144 | 0.135 | 0.135 | 0.138 | 0.144 |
| credit-g.arff | 0.207 | 0.215 | 0.221 | 0.219 | 0.207 | 0.215 | 0.221 | 0.219 |
| diabetes.arff | 0.198 | 0.200 | 0.202 | 0.204 | 0.196 | 0.199 | 0.205 | 0.204 |
| glass2.arff | 0.100 | 0.095 | 0.094 | 0.093 | 0.101 | 0.100 | 0.108 | 0.103 |
| hepatitis.arff | 0.138 | 0.150 | 0.148 | 0.146 | 0.138 | 0.150 | 0.148 | 0.146 |
| hypothyroid.arff | 0.039 | 0.040 | 0.040 | 0.041 | 0.039 | 0.040 | 0.040 | 0.041 |
| ionosphere.arff | 0.068 | 0.069 | 0.069 | 0.069 | 0.073 | 0.067 | 0.070 | 0.069 |
| kr-vs-kp.arff | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 | 0.007 |
| labor.arff | 0.072 | 0.065 | 0.062 | 0.058 | 0.072 | 0.065 | 0.062 | 0.058 |
| lymph.arff | 0.123 | 0.136 | 0.141 | 0.133 | 0.123 | 0.136 | 0.141 | 0.133 |
| segment.arff | 0.027 | 0.028 | 0.029 | 0.029 | 0.027 | 0.028 | 0.029 | 0.029 |
| sick.arff | 0.052 | 0.053 | 0.053 | 0.054 | 0.052 | 0.053 | 0.053 | 0.054 |
| solar-flare-1.arff | 0.023 | 0.027 | 0.026 | 0.027 | 0.023 | 0.027 | 0.027 | 0.027 |
| sonar.arff | 0.153 | 0.172 | 0.177 | 0.171 | 0.149 | 0.177 | 0.177 | 0.162 |
| soybean.arff | 0.052 | 0.052 | 0.052 | 0.054 | 0.050 | 0.052 | 0.058 | 0.056 |
| sponge.arff | 0.051 | 0.053 | 0.053 | 0.053 | 0.051 | 0.053 | 0.053 | 0.053 |
| vote.arff | 0.034 | 0.033 | 0.033 | 0.032 | 0.034 | 0.033 | 0.033 | 0.032 |
| vowel.arff | 0.049 | 0.054 | 0.053 | 0.055 | 0.045 | 0.048 | 0.048 | 0.049 |
| zoo.arff | 0.031 | 0.032 | 0.031 | 0.036 | 0.031 | 0.032 | 0.031 | 0.036 |
| Average | 0.089 | 0.092 | 0.093 | 0.093 | 0.090 | 0.092 | 0.094 | 0.092 |

Table 17: Detailed Variance for RF ensembles M=1, 3; and Trees=10, 50, 100, 200; ($RF^{M}_{Trees}$) - Section 4.4.4

| Dataset | $RF^{1}_{10}$ | $RF^{1}_{50}$ | $RF^{1}_{100}$ | $RF^{1}_{200}$ | $RF^{3}_{10}$ | $RF^{3}_{50}$ | $RF^{3}_{100}$ | $RF^{3}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.025 | 0.013 | 0.011 | 0.011 | 0.019 | 0.012 | 0.011 | 0.011 |
| audiology.arff | 0.224 | 0.152 | 0.143 | 0.138 | 0.233 | 0.155 | 0.139 | 0.141 |
| autos.arff | 0.173 | 0.134 | 0.130 | 0.127 | 0.171 | 0.135 | 0.140 | 0.135 |
| breast-cancer.arff | 0.073 | 0.062 | 0.062 | 0.059 | 0.087 | 0.081 | 0.083 | 0.083 |
| colic.arff | 0.093 | 0.046 | 0.040 | 0.035 | 0.080 | 0.040 | 0.037 | 0.030 |
| credit-g.arff | 0.066 | 0.030 | 0.026 | 0.025 | 0.071 | 0.041 | 0.038 | 0.037 |
| diabetes.arff | 0.088 | 0.068 | 0.063 | 0.062 | 0.085 | 0.067 | 0.068 | 0.062 |
| glass2.arff | 0.126 | 0.081 | 0.088 | 0.081 | 0.121 | 0.098 | 0.087 | 0.083 |
| hepatitis.arff | 0.047 | 0.026 | 0.030 | 0.034 | 0.052 | 0.039 | 0.035 | 0.036 |
| hypothyroid.arff | 0.014 | 0.010 | 0.009 | 0.009 | 0.015 | 0.011 | 0.011 | 0.010 |
| ionosphere.arff | 0.031 | 0.016 | 0.012 | 0.009 | 0.034 | 0.017 | 0.014 | 0.014 |
| kr-vs-kp.arff | 0.046 | 0.020 | 0.018 | 0.015 | 0.021 | 0.011 | 0.009 | 0.008 |
| labor.arff | 0.118 | 0.060 | 0.068 | 0.068 | 0.089 | 0.066 | 0.062 | 0.052 |
| lymph.arff | 0.097 | 0.038 | 0.040 | 0.033 | 0.072 | 0.043 | 0.039 | 0.036 |
| segment.arff | 0.047 | 0.028 | 0.025 | 0.024 | 0.041 | 0.025 | 0.023 | 0.021 |
| sick.arff | 0.011 | 0.009 | 0.008 | 0.008 | 0.012 | 0.010 | 0.009 | 0.010 |
| solar-flare-1.arff | 0.004 | 0.006 | 0.004 | 0.004 | 0.007 | 0.005 | 0.004 | 0.005 |
| sonar.arff | 0.174 | 0.110 | 0.082 | 0.076 | 0.150 | 0.087 | 0.073 | 0.071 |
| soybean.arff | 0.066 | 0.030 | 0.025 | 0.022 | 0.052 | 0.026 | 0.023 | 0.021 |
| sponge.arff | 0.012 | 0.011 | 0.011 | 0.011 | 0.012 | 0.011 | 0.011 | 0.011 |
| vote.arff | 0.027 | 0.016 | 0.012 | 0.010 | 0.018 | 0.008 | 0.007 | 0.007 |
| vowel.arff | 0.207 | 0.128 | 0.117 | 0.114 | 0.177 | 0.124 | 0.117 | 0.113 |
| zoo.arff | 0.035 | 0.039 | 0.032 | 0.032 | 0.049 | 0.046 | 0.033 | 0.042 |
| Average | 0.078 | 0.049 | 0.046 | 0.044 | 0.073 | 0.050 | 0.047 | 0.045 |

Table 18: Detailed Variance for RF ensembles M=5, Log N; and Trees=10, 50, 100, 200; ($RF^{M}_{Trees}$) - Section 4.4.4

| Dataset | $RF^{5}_{10}$ | $RF^{5}_{50}$ | $RF^{5}_{100}$ | $RF^{5}_{200}$ | $RF^{LogN}_{10}$ | $RF^{LogN}_{50}$ | $RF^{LogN}_{100}$ | $RF^{LogN}_{200}$ |
|---|---|---|---|---|---|---|---|---|
| anneal.arff | 0.016 | 0.010 | 0.008 | 0.008 | 0.013 | 0.010 | 0.009 | 0.008 |
| audiology.arff | 0.212 | 0.148 | 0.145 | 0.139 | 0.201 | 0.150 | 0.139 | 0.139 |
| autos.arff | 0.173 | 0.149 | 0.145 | 0.148 | 0.173 | 0.149 | 0.145 | 0.148 |
| breast-cancer.arff | 0.100 | 0.100 | 0.093 | 0.093 | 0.094 | 0.086 | 0.086 | 0.090 |
| colic.arff | 0.055 | 0.034 | 0.030 | 0.027 | 0.055 | 0.034 | 0.030 | 0.027 |
| credit-g.arff | 0.077 | 0.053 | 0.045 | 0.045 | 0.077 | 0.053 | 0.045 | 0.045 |
| diabetes.arff | 0.087 | 0.072 | 0.067 | 0.065 | 0.081 | 0.067 | 0.063 | 0.062 |
| glass2.arff | 0.110 | 0.095 | 0.092 | 0.094 | 0.111 | 0.090 | 0.078 | 0.079 |
| hepatitis.arff | 0.067 | 0.038 | 0.033 | 0.037 | 0.067 | 0.038 | 0.033 | 0.037 |
| hypothyroid.arff | 0.016 | 0.012 | 0.012 | 0.012 | 0.016 | 0.012 | 0.012 | 0.012 |
| ionosphere.arff | 0.034 | 0.019 | 0.017 | 0.017 | 0.032 | 0.022 | 0.019 | 0.019 |
| kr-vs-kp.arff | 0.013 | 0.006 | 0.006 | 0.006 | 0.012 | 0.007 | 0.006 | 0.006 |
| labor.arff | 0.074 | 0.068 | 0.064 | 0.068 | 0.074 | 0.068 | 0.064 | 0.068 |
| lymph.arff | 0.081 | 0.047 | 0.035 | 0.041 | 0.081 | 0.047 | 0.035 | 0.041 |
| segment.arff | 0.034 | 0.022 | 0.021 | 0.020 | 0.034 | 0.022 | 0.021 | 0.020 |
| sick.arff | 0.013 | 0.011 | 0.011 | 0.010 | 0.013 | 0.011 | 0.011 | 0.010 |
| solar-flare-1.arff | 0.008 | 0.004 | 0.006 | 0.005 | 0.008 | 0.005 | 0.003 | 0.004 |
| sonar.arff | 0.141 | 0.079 | 0.057 | 0.058 | 0.147 | 0.074 | 0.067 | 0.071 |
| soybean.arff | 0.048 | 0.030 | 0.026 | 0.024 | 0.055 | 0.029 | 0.024 | 0.025 |
| sponge.arff | 0.011 | 0.011 | 0.011 | 0.011 | 0.014 | 0.011 | 0.011 | 0.011 |
| vote.arff | 0.014 | 0.007 | 0.007 | 0.006 | 0.014 | 0.007 | 0.007 | 0.006 |
| vowel.arff | 0.175 | 0.135 | 0.130 | 0.127 | 0.177 | 0.126 | 0.119 | 0.120 |
| zoo.arff | 0.051 | 0.046 | 0.042 | 0.038 | 0.051 | 0.046 | 0.042 | 0.038 |
| Average | 0.070 | 0.052 | 0.048 | 0.048 | 0.070 | 0.051 | 0.046 | 0.047 |

# References


[1] *http://www.ncbi.nlm.nih.gov/*. 100

[2] J. Abellán. Application of uncertainty measures on credal sets on the naive bayes classifier. *International J. of General System*, 35:675–686, 2006. 34

[3] J. Abellán and A. R. Masegosa. Combining decision trees based on imprecise probabilities and uncertainty measures. In K. Mellouli, editor, *ECSQARU*, volume 4724 of *Lecture Notes in Computer Science*, pages 512–523. Springer, 2007. 84

[4] D. W. Aha and R. L. Bankert. Feature selection for case-based classification of cloud types: An empirical comparision. In *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning*, page 106112, Seattle, WA, 1994. AAAI Press. 127, 128

[5] A. Alizadeh, M. Eisen, E. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, J. H. Jr, L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Botstein, P. Brown, and L. Staudt. Distinct types of diffuse large B–cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, February 2000. xii, 106, 110, 111, 120, 121, 129

[6] H. Allmuallim and T. Dietterich. Learning with many irrelevant features. In *Ninth National Conference on Artificial Intelligence*, pages 547–552. MIT Press, 1991. 127

[7] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96:6745–6750, 1999. 106

[8] T. Ando, M. Katayama, M. Seto, T. Kobayashi, and H. Honda. Selection of causal gene sets from transciptional profiling by fnn modeling an prediction of lymphoma outcome. *Gene Informatics*, 13:278–279, 2002. 121, 132

[9] J. Bai, J.-Y. Nie, G. Cao, and H. Bouchard. Using query contexts in information retrieval. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 15–22, New York, NY, USA, 2007. ACM. 139

[10] C. Barry and L. Schamber. Users' Criteria for Relevance Evaluation: A Cross-situational Comparison. *Information Processing and Management*, 34(2-3):219–236, 1998. 146, 172

[11] N. J. Belkin. Intelligent information retrieval: Whose intelligence? In *ISI '96: Proceedings of the Fifth International Symposium for Information Science*, pages 25–31, 1996. 143

[12] A. Ben-Dor, N. Friedman, and Z. Yakhini. Scoring genes for relevance. Technical Report 2000-38, School of Computer Science & Engineering, Hebrew University, Jerusalem, 2000. 107

[13] A. Bernard and A. J. Hartemink. Informative structure priors: joint learning of dynamic regulatory networks from multiple types of data. *Pac Symp Biocomput*, pages 459–470, 2005. 106

[14] K. Bharat. Searchpad: explicit capture of search context to support web search. In *Proceedings of the 9th international World Wide Web conference on Computer networks : the international journal of computer and telecommunications netowrking*, pages 493–501, Amsterdam, The Netherlands, The Netherlands, May 2000. North-Holland Publishing Co. 141, 142

[15] R. Blanco and A. Barreiro. Probabilistic document length priors for language models. pages 394–405. 2008. 153

[16] R. Blanco, P. Larrañaga, I. Inza, and B. Sierra. Selection of highly accurate genes for cancer classification by estimation of distribution algorithms. In *Workshop of Bayesian Models in Medicine, AIME01*, pages 29–34, 2001. 107

[17] J. Bockhorst, M. Craven, D. Page, J. Shavlik, and J. Glasner. A Bayesian network approach to operon prediction. *Bioinformatics*, 19(10):1227–1235, 2003. 107

[18] C. E. Bonferroni. Il calcolo delle assicurazioni su gruppi di teste. *Studi in Onore del Professore Salvatore Ortu Carboni*, pages 13–60, 1935. 161

[19] P. Borlund. Experimental components for the evaluation of interactive information retrieval systems. *Journal of Documentation*, 56(1):71–90, 2000. 164

[20] P. Borlund. The iir evaluation model: a framework for evaluation of interactive information retrieval systems. *Information Research*, 8(3), 2003. Paper No. 152 [Available at http://informationr.net/ir/8-3/paper152.html]. 139

[21] C. Boutilier, N. Friedman, M. Goldszmidt, and D. Koller. Context-specific independence in Bayesian networks. *Uncertainty in Artificial Intelligence. Proceedings of the 20th Conference*, pages 115–123, 1996. 156

[22] D. Bowser-Chao and L. Debra. A comparison of the use of binary decision trees and neural networks in top quark detection. *Physical Review D: Particles and Fields 47, pp.1900-1905*, 1993. 19

[23] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996. 2, 22, 23, 34, 73, 82

[24] L. Breiman. Out-of-bag estimation. *Private communication*, 1996. 22, 73

[25] L. Breiman. Arcing classifiers. *The Annals of Statistics*, 26(3):801–824, 1998. 22, 83

[26] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. 4, 22, 23, 24, 25, 67, 82, 83, 85, 89, 93

[27] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. Wadsworth International Group, Belmont, 1984. 2, 20, 21, 22, 71, 82, 83, 85

[28] P. J. Brown, J. D. Bovey, and X. Chen. Context-aware applications: from the laboratory to the marketplace. *IEEE Personal Communications*, 4(5):58–64, October 1997. 141

[29] J. Budzik and K. J. Hammond. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*, New Orleans, Louisiana, 2000. ACM Press. 141

[30] W. Buntine. Learning classification trees. *Statistics and Computing*, (2):63–73, 1992. 68, 70, 71, 72

[31] I. Campbell. *The ostensive model of developing information needs*. Ph.d. thesis, University of Glasow, Glasgow, UK, 2000. xii, 157

[32] I. Campbell and K. van Rijsbergen. The ostensive model of developing information needs. In P. Ingwersen and N. Ole Pors, editors, *Proceedings of the Second International Conference on Conceptions of Library and Information Science*, pages 251–268, Copenhagen, Denmark, 1996. 141

[33] R. Caruana, N. Karampatziakis, and A. Yessenalina. An empirical evaluation of supervised learning in high dimensions. In *ICML '08: Proceedings of the 25th international conference on Machine learning*, pages 96–103, New York, NY, USA, 2008. ACM. 33

[34] R. Caruana and A. Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd International Conference on Machine Learning (ICML'06)*, 2006. 33

[35] J. H. Chang, K. B. Hwang, and B. T. Zhang. Analysis of gene expression profiles and drug activity patterns by clustering and Bayesian network learning. In *Methods of Microarray Data Analysis II (CAMDA'01)*, pages 169–184, 2002. 107

[36] Y.-P. Chen, J.-Y. Yang, S.-N. Liou, G.-Y. Lee, and J.-S. Wanga. Online classifier construction algorithm for human activity detection using a tri-axial accelerometer. *Applied Mathematics and Computation*, 205:849–860, 2008. 33

[37] C. K. Chow and C. N. Liu. Approximating discrete probability distributions. *IEEE Transactions on Information Theory, 14*, pages 462–467, 1968. 16

[38] P. R. Christopher D. Manning and H. Schtze. *Introduction to Information Retrieval,*. Cambridge University Press, 2008. 137

[39] E. Consortium. Elvira: An environment for probabilistic graphical models. In J. Gámez and A. Salmerón, editors, *Proceedings of the 1st European Workshop on Probabilistic Graphical Models*, pages 222–230, 2002. 46, 47, 60, 77, 87

[40] C. Cool and A. Spink. Issues of context in information retrieval (ir): an introduction to the special issue. *Inf. Process. Manage.*, 38(5):605–611, 2002. 139, 146

[41] G. Cooper and E. Herskovits. A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992. 119

[42] R. Cowell, A. Dawid, S. Lauritzen, and D. Spiegelhalter. *Probabilistic Networks and Expert Systems*. Statistics for Engineering and Information Science.Springer-Verlag, New York, 1999. 10, 114

[43] N. Craswell, S. Robertson, H. Zaragoza, and M. Taylor. Relevance weighting for query independent evidence. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development*

*in information retrieval*, pages 416–423, New York, NY, USA, 2005. ACM Press. 153, 154

[44] F. Crestani and I. Ruthven, editors. *Information Context: Nature, Impact, and Role; 5th International Conference on Conceptions of Library and Information Sciences, CoLIS 2005, Glasgow, UK, June 4-8, 2005; Proceedings*, volume 3507 of *Lecture Notes in Computer Science*. Springer, 2005. 146

[45] F. Crestani and I. Ruthven. Introduction to special issue on contextual information retrieval systems. *Information Retrieval*, 10(2):111–113, April 2007. 146

[46] B. de Finetti. *Theory of Probability*. J. Wiley and Sons, Inc, New York, 1974. 158

[47] J. Demsar. Statistical comparisons of classifiers over multiple data sets. *Journal Machine Learning Research*, 7:1–30, 2006. 61, 78, 88, 92, 93

[48] T. G. Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 2000. 23, 24, 82

[49] P. Domingos. Bayesian model averaging in rule induction. In *In Preliminary papers of the Sixth International Workshop on Artificial Intelligence and Statistics*, pages 157–164, 1997. 72

[50] P. Domingos and M. J. Pazzani. On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, 29(2-3):103–130, 1997. 9, 14, 15

[51] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continious features. In *Twelfth International Conference on Machine Learning*, pages 194–202, 1995. 10

[52] P. Dourish. What we talk about when we talk about context. *Personal and Ubiquitous Computing*, 8(1):19–30, 2004. 140, 179

[53] S. Draghici, O. Kulaeva, B. Hoff, A. Petrov, S. Shams, and M. Tainsky. Noise sampling method: an anova approach allowing robust selection of differentially regulated genes measured by dna microarrays. *Bioinformatics*, 19:1348–1359, 2003. 116

[54] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley Sons, New York, 1973. 2, 8, 9, 56, 114

[55] R. O. Duda and P. E. Hart. *Pattern Classification*. Wiley Interscience, 2000. 160

[56] B. Edmonds. The pragmatic roots of context. In P. Bouquet, L. Serafini, P. Brézillon, M. Benerecetti, and F. Castellani, editors, *CONTEXT*, volume 1688 of *Lecture Notes in Computer Science*, pages 119–132. Springer, 1999. 141

[57] J. A. Falconer, B. J. Naughton, D. D. Dunlop, E. J. Roth, and D. Strasser. Predicting stroke inpatient rehabilitation outcome using a classification tree approach. *Archives of Physical Medicine and Rehabilitation, 75(6):619*, 1994. 19

[58] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of 13th International Joint Conference on AI*, 1993. 47, 61, 88

[59] M. Feld and G. Kahl. Integrated speaker classification for mobile shopping applications. In *AH*, pages 288–291, 2008. 33

[60] P. E. File, P. I. Dugard, and A. S. Houston. Evaluation of the use of induction in the develeopment of a medical expert system. *Computers and Biomedical Research, 27(5):383–395*, 1994. 19

[61] L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. Placing search in context: the concept revisited. In *WWW '01: Proceedings of the 10th international conference on World Wide Web*, pages 406–414, New York, NY, USA, 2001. ACM. 139, 141

[62] S. Fox, K. Karnawat, M. Mydland, S. Dumais, and T. White. Evaluating implicit measures to improve web search. *ACM Transactions on Information Systems*, 23(2):147–168, 2005. 1059982. 139, 147, 179

[63] L. Freund, E. G. Toms, and C. L. A. Clarke. Modeling task-genre relationships for IR in the workspace. In *Proceedings of the 28th SIGIR Conference*, pages 441–448, Salvador, Brazil, 2005. ACM. 140, 147, 148, 178

[64] Y. Freund and R. E. Schapire. A decision-theoretic generalization of online learning and an application to boosting. In *European Conference on Computational Learning Theory*, pages 23–37, 1995. 2, 22, 25, 34, 67

[65] N. Friedman. The Bayesian structural em algorithm. In *14th Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 129–138, 1998. 107

[66] N. Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659):799–805, 2004. 106

[67] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, 1997. 56

[68] N. Friedman and M. Goldszmidt. Learning Bayesian networks with local structure. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)*, pages 252–262, Portland, Oregon, 1996. 3, 15

[69] N. Friedman, M. Goldszmidt, and A. J. Wyner. Data analysis with Bayesian networks: A bootstrap approach. In *15th Conference on Uncertainty in Artificial Intelligence (UAI–99)*, pages 196–205, 1999. 106

[70] N. Friedman, M. Linial, I. Nachman, and D. Peér. Using bayesian networks to analyze expression data. *Journal of Computational Biology*, 7:601–620, 2000. 106

[71] D. Geiger and D. Heckerman. Knowledge representation and inference in similarity networks and Bayesian multinets. *Artificial Intelligence, 82*, pages 45–74, 1996. 156

[72] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, April 2006. 22, 25, 83, 93

[73] P. Geurts and L. Wehenkel. Investigation and reduction of discretization variance in decision tree induction. In *Proceedings of the 11th European Conference on Machine Learning (ECML-2000)*, pages 162–170. Springer Verlag, 2000. 83

[74] D. K. Gifford. Blazing pathways through genetic mountains. *Science*, (293):2049–2051, 2001. 106

[75] E. J. Glover, S. Lawrence, W. P. Birmingham, and L. C. Giles. Architecture of a metasearch engine that supports user information needs. In *Eighth International Conference on Information and Knowledge Management (CIKM'99)*, pages 210–216, Kansas City, MO, November 1999. ACM Press. 141

[76] T. Gneiting and A. E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Technical Report. Department of Statistics, University of Washington*, 463R, 2005. 78

[77] M. A. Hall. Correlation-based feature selection for discrete and numeric class machine learning. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 359–366, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc. 28

[78] M. A. Hall and G. Holmes. Benchmarking attribute selection techniques for discrete class data mining. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):1437–1447, November/December 2003. 160

[79] A. J. Hartemink et al. Combining location and expression data for principled discovery of genetic regulatory network models. *Pac Symp Biocomput*, pages 437–449, 2002. 106

[80] D. Heckerman, D. Geiger, and D. M. Chickering. Learning bayesian networks: The combination of knowledge and statistical data. In *KDD Workshop*, pages 85–96, 1994. 36, 37, 76

[81] P. Helman, R. Veroff, S. R. Atlas, and C. Willman. A Bayesian network classification methodology for gene expression data. *Journal of Computational Biology*, 11(4):581 – 615, 2004. 107

[82] M. Henrion. Propagating uncertainty by logic sampling in bayes' networks. In J. Lemmer and L. Kanal, editors, *Uncertainty in Artificial Intelligence, 2*, pages 149–164. Amsterdam, 1988. 46

[83] J. A. Hoeting, D. Madigan, A. E. Raftery, and C. T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–417, 1999. 68

[84] C.-N. Hsu, H.-J. Huang, and T.-T. Wong. Why discretization works for naïve bayesian classifiers. In *Proceedings 17th International Conference on Machine Learning*, pages 399–406. Morgan Kaufmann, San Francisco, CA, 2000. 114

[85] E. Hun, J. Marin, and P. Stone. Experiments in induction. *Academic Press*, 1966. 19

[86] D. Husmeier. Reverse engineering of genetic networks with Bayesian networks. *Biochem Soc Trans*, 31(Pt 6):1516–1518, 2003. 106

[87] K. B. Hwang et al. Applying machine learning techniques to analysis of gene expression data: Cancer diagnosis. In *Methods of Microarray Data Analysis (CAMDA'00)*, pages 167–182, 2002. 107

[88] S. Imoto et al. Bayesian network and nonparametric heteroscedastic regression for nonlinear modeling of genetic network. In *IEEE Computer Society Conference on Bioinformatics (CSB '02)*, page 219, 2002. 106

[89] P. Ingwersen. Cognitive perspectives of information retrieval interaction: elements of a cognitive ir theory. *Journal of Documentation*, 52:3–50, 1996. 143

[90] P. Ingwersen and N. Belkin. Information retrieval in context - IRiX: workshop at SIGIR 2004. *SIGIR Forum*, 38(2):50–52, 2004. 146

[91] P. Ingwersen and K. Järvelin. Information retrieval in context: IRiX. *SIGIR Forum*, 39(2):31–39, 2005. 144, 146

[92] P. Ingwersen and K. Järvelin. *The Turn: Integration of Information Seeking and Retrieval in Context*. Springer, 2006. xii, 139, 140, 146, 177, 179

[93] I. Inza, P. Larrañaga, R. Blanco, and A. J. Cerrolaza. Filter versus wrapper gene selection approaches in dna microarray domains. *Artificial Intelligence in Medicine*, 31(2):91–103, 2004. 107

[94] I. Inza, B. Sierra, R. Blanco, and P. Larrañaga. Gene selection by sequential search wrapper approaches in microarray cancer class prediction. *Journal of Intelligent and Fuzzy Systems*, 12(1):25–34, 2002. 107

[95] M. Jaeger, J. D. Nielsen, and T. Silander. Learning probabilistic decision graphs. *International Journal of Approximate Reasoning*, 42:84–100, 2006. 156

[96] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002. 158

[97] L. Jiang and H. Zhang. Weightily averaged one-dependence estimators. In *PRICAI*, pages 970–974, 2006. 34, 57

[98] G. H. John and R. Kohavi. Irrelevant features and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 121–129, 1994. 127

[99] G. H. John and P. Langley. Estimating continuous distributions in Bayesian classifiers. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 338–345. Morgan Kaufmann Publishers, San Mateo, 1995. 10, 114

[100] H. Joho and J. M. Jose. Slicing and dicing the information space using local contexts. In *Proceedings of the First Symposium on Information Interaction in Context (IIiX)*, pages 111–126, Copenhagen, Denmark, 2006. 163

[101] H. Joho and J. M. Jose. Effectiveness of additional representations for the search result presentation on the web. *Information Processing & Management*, page to appear, 2007. 163

[102] M. Jones, G. Buchanan, and H. W. Thimbleby. Sorting out searching on small screen devices. In F. Paternò, editor, *4th International Symposium on Mobile Human-Computer Interaction*, Lecture Notes in Computer Science, Vol. 2411, pages 81–94, Pisa, Italy, 2002. Springer. 140

[103] G. Judmaier, P. Meyersbach, G. Weiss, H. Wachter, and G. Reibnegger. The role of neopterin in assessing disease activity in crohn's disease: Classification and regression trees. *The American Journal of Gastroenterology, 88: 706-711*, 1993. 19

[104] D. Kelly. Measuring online information seeking context, part 1: Background and method. *J. Am. Soc. Inf. Sci. Technol.*, 57(13):1729–1739, 2006. 139

[105] D. Kelly and N. J. Belkin. Display time as implicit feedback: understanding task effects. In *Proceedings of the 27th SIGIR Conference*, pages 377–384, Sheffield, UK, 2004. ACM Press. 146

[106] D. Kelly and J. Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003. 959260. 141

[107] S. Y. Kim, S. Imoto, and S. Miyano. Inferring gene networks from time series microarray data using dynamic Bayesian networks. *Brief Bioinform*, 4(3):228–235, 2003. 106

[108] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In N. Lavrac and S. Wrobel, editors, *Proccedings of the International Joint Conference on Artificial Intelligence*, 1995. 30, 118, 121

[109] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997. 29, 30, 58, 127, 128, 160

[110] R. Kohavi and D. Wolpert. Bias plus variance decomposition for zero-one loss functions. In *ICML*, pages 275–283, 1996. 83, 88

[111] P. Kokol, M. Mernik, J. Zavrsnik, and K. Kancler. Decision trees based on automatic learning and their use in cardiology. *Journal of Medical Systems, 18(4):201*, 1994. 19

[112] I. Kononenko. Semi-naive Bayesian classifier. In *European working session on learning on Machine learning*, pages 206–219, 1991. 14, 36, 58

[113] I. Kononenko. Inductive and Bayesian learning in medical diagnosis. *Applied Artificial Intelligence 7(4): 317-337*, 1993. 19

[114] W. Kraaij. The importance of prior probabilities for entry page search. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 27–34. ACM Press, 2002. 153, 154

[115] R. Kuppers. Mechanisms of b-cell lymphoma pathogenesis. *Nature Reviews Cancer*, 5:251–262, April 2005. xii, 109

[116] J. M. Lachin. On a stepwise procedure for two population bayes decision rules using discrete variables. *Biometrics*, 29(3):551–564, 1973. 162

[117] J. Lafferty and C. Zhai. Probabilistic relevance models based on document and query generation. In *Language Modeling and Information Retrieval*, pages 1–10. Kluwer Academic Publishers, 2003. 153

[118] P. Langley, W. Iba, and K. Thompson. An analysis of bayesian classifiers. In *National Conference on Artificial Intelligence*, pages 223–228, 1992. 9, 115

[119] P. Langley and S. Sage. Induction of selective bayesian classifiers. *Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence*, pages 399–406, 1994. 11, 114

[120] P. Langley and S. Sage. Oblivious decision trees and abstract cases. In *Working Notes of the AAAI-94 Workshop on Case-Based Reasoning*, Seattle, WA, 1994. AAAI Press. 127, 128

[121] V. Lantz and R. MurraySmith. Rhythmic interaction with a mobile device. *NordiCHI*, 4:97–100, 2004. 33

[122] P. Larrañaga and J. A. Lozano. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation*. Genetic Algorithms and Evolutionary Computation. Kluwer Academic Publishers, Norwell, MA, USA, 2001. 107

[123] S. Lawrence. Context in web search. *IEEE Data Engineering Bulletin*, 23(3):25–32, 2000. 141, 142

[124] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method. *Bioinformatics*, 17:1131–1142, 2001. 121, 132

[125] I. S. Lossos. Prediction of survival in diffuse large-b-cell lymphoma based on the expression of six genes. *The New England Journal of Medicine*, 350:1828–1837, 2004. 116, 123

[126] A. R. Masegosa, H. Joho, and J. M. Jose. Evaluating query-independent object features for relevancy prediction. In *Proceedings of ECIR 2007*, pages 283–294, Rome, Italy, 2007. 155

[127] R. M. A. Mateo, L. F. Cervantes, H.-K. Yang, and J. Lee. Mobile agents using data mining for diagnosis support in ubiquitous healthcare. In *KES-AMSTA*, pages 795–804, 2007. 33

[128] D. McKenzie, P. McGorry, C. Wallace, L. H. Low, D. Copolov, and B. Singh. Constructing a minimal diagnostic decision tree. *Methods of Information in Medicine, 32(2):161–166*, 1993. 19

[129] J. Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3(4):319–342, 1989. 23, 82

[130] T. Mitchell. *Machine Learning*. MacGraw-Hill Companies, 1997. 10

[131] E. Moler, M. Chow, and I. Mian. Analysis of molecular profile data using generative and discriminative methods. *Physiological Genomics*, 4(2):109–126, 2000. 107

[132] A. W. Moore and M. Lee. Efficient algorithms for minimizing cross validation error. In *Eleventh International Conference on Machine Learning*. San Francisco: Morgan Kaufmann, 1994. 30

[133] K. Murphy and S. Mian. Modelling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, Univ. of California, 1999. 106

[134] C. Nadeau and Y. Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003. 47, 161

[135] C. Nadeau and Y. Bengio. Inference for the generalization error. *Machine Learning*, 52(3):239–281, 2003. 169

[136] R. E. Neapolitan. *Learning Bayesian Networks*. Prentice Hall, 2004. 38

[137] M. J. Pazzani. Searching for dependencies in bayesian classifiers. *Lecture Notes in Statistics*, 112:239–248, 1995. 12, 13, 14, 15, 29, 30, 36, 58, 60

[138] M. J. Pazzani, C. J. Merz, P. M. Murphy, K. Ali, T. Hume, and C. Brunk. Reducing misclassification costs. In *ICML*, pages 217–225, 1994. 73

[139] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17 Suppl 1:S215–224, 2001. 106

[140] J. M. Peña, J. A. Lozano, and P. Larrañaga. Unsupervised learning of Bayesian networks via estimation of distribution algorithms: an application to gene expression data clustering. *International J. Uncertain. Fuzziness Knowl.-Based Syst.*, 12(SUPPLEMENT):63–82, 2004. 107

[141] C. I. Peng, J.;Macdonald. Automatic document prior feature selection for web retrieval. In *In Proceedings of the 31st Annual International ACM SIGIR Conference (SIGIR 2008), 20-24 July 2008, Singapore*. ACM, 2008. 154, 155

[142] I. Peng, J.;Ounis. Combination of document priors in web information retrieval. In *Proceedings of the 29th European Conference on Information Retrieval (ECIR), Rome, Italy, 2007*. Poster paper. 153

[143] B.-E. Perrin, L. Ralaivola, A. Mazurie, S. Bottani, J. Mallet, and F. D'Alch-Buc. Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19 Suppl 2:II138–II148, 2003. 106

[144] S. Pomeroy, P. Tamayo, M. Gaasenbeek, L. Sturla, M. Angelo, M. McLaughlin, J. Kim, L. Goumnerova, P. Black, C. Lau, J. Allen, D. Zagzag, J. Olson, T. Curran, C. Wetmore, J. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. Louis, J. Mesirov, E. Lander, and T. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415:436–442, 2002. 106

[145] J. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *Proceedings of the ACM SIGIR*, pages 275–281, 1998. 151

[146] F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, 2003. 73

[147] J. Quinlan. Discovering rules by induction from large collection of examples. *Knowledge-base systems in the Micro Electronic Age, Edinburgh University Press, pp. 168-201*, 1979. 19

[148] J. Quinlan. *C4.5: programs for machine learning*. San Francisco: Morgan Kaufmann, 1993. 2, 19, 71

[149] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco, 1993. 20, 34, 59, 60, 73, 83

[150] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986. 20, 71, 85

[151] S. E. Robertson. The probability ranking principle in ir. pages 281–286, 1997. 73

[152] A. Rosenwald, G. Wright, W. C. Chan, J. M. Connors, E. Campo, R. I. Fisher, R. D. Gascoyne, H. K. Muller-Hermelink, E. B. Smealand, and L. M. Staudt. The use of molecular profiling to predict survival after chemotherapy for diffuse large-b-cell lymphoma. *New England Journal of Medicine*, 346:1937–1947, June 2002. 122, 132

[153] M. S. Roulston and L. Smith. Evaluating probabilistic forecasts using information theory. *Monthly Weather Review*, 130:1653–1660, 2002. 78, 119

[154] I. Ruthven, P. Borlund, P. Ingwersen, N. Belkin, A. Tombros, and P. Vakkari, editors. *Proceedings of the 1st IIiX Symposium*, Copenhagen, Denmark, 2006. 146

[155] M. Sahami. Learning limited dependence Bayesian classifiers. *Second International Conference on Knowledge Discovery in Databases*, pages 335–338, 1996. 3, 16, 17

[156] S. Salzberg, R. Chandar, H. Forf, S. Murth, and R. White. Decision trees for automated identification of cosmic-ray hits in hubble space telescope images. *Publications of the Astronomical Society of the Pacific, 107:1–10*, 1995. 19

[157] T. Saracevic. Interactive models in information retrieval: a review and proposal. In *Proceedings of the 59th Annual Meeting of the American Society for Information Science*, pages 3–9, 1996. 142

[158] R. E. Schapire. The strength of weak learnability. pages 197–227, 1990. 25

[159] B. Seroussi and J. L. Golmard. An algorithm directly finding the k most probable configurations in bayesian networks. *International Journal of Approximate Reasoning*, 11:205–233, 1994. 119

[160] R. Shachter and C. Kenley. Gaussian influence diagrams. *Management Science*, 35:527–550, 1989. 10

[161] C. Shannon and W. Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, Urbana, Illinois, 1949. 150

[162] C. E. Shannon. A mathematical theory of communication. *Bell System Tech. J. 27, 379-423, 623-656*, 1948. 20

[163] S. Shimozono, A. Shinohara, T. Shinohara, S. Miyano, S. Kuhara, and S. Arikawa. Knowledge acquisition from amino acid sequences by machine learning system bonsai. *Transactions on Information Processing Society of Japan, Vol.35, No.10, pp.2009-2018*, 1994. 19

[164] E. Southern. Detection of specific sequences among dna fragments separated by gel electrophoresis. *Journal of Molecular Biology*, 68:503–517, 1975. 102

[165] A. Spink. Term relevance feedback and query expansion: Relation to design. In B. W. Croft and C. J. Van Rijsbergen, editors, *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 81–90, Berlin, Germany, 1994. ACM. 143

[166] A. Spink. Study of interactive feedback during mediated information retrieval. *Journal of the American Society for Information Science and Technology*, 5(48):382–394, 1997. 143

[167] P. Spirtes et al. Constructing Bayesian network models of gene expression networks from microarray data. In *Proceedings of the Atlantic Symposium on Computational Biology, Genome Information Systems and Technology*, 2000. 106

[168] M. Stone. An asymptotic equivalence of choice of model by cross-validation and akaike's criterion. *Journal of the Royal Statistical Society*, 38:48–47, 1997. 29, 115, 129

[169] Y. Tamada et al. Estimating gene networks from gene expression data by combining Bayesian network model with promoter element detection. *Bioinformatics*, 19 Suppl 2:II227–II236, 2003. 106

[170] B. Thiesson, C. Meek, D. M. Chickerin, and D. Heckerman. Learning mixtures of DAG models. *Proceeding of the 14th Conference on Uncertainty in Artificial Intelligence*, pages 504–513, 1998. 156

[171] P. Thompson. Subjective probability and information retrieval: a review of the psychological literature. 44:119–143, 1988. 158

[172] A. Tombros, I. Ruthven, and J. M. Jose. How users assess web pages for information seeking. *Journal of the American Society for Information Science and Technology*, 56(4):327–344, 2005. 146, 147, 178

[173] L. J. vant Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002. 106

[174] L. Wasserman. Bayesian model selection and model averaging. *J. Math. Psychol.*, 44(1):92–107, 2000. 68

[175] L. Wasserman. *All of Statistics : A Concise Course in Statistical Inference (Springer Texts in Statistics)*. Springer, September 2004. 40, 44

[176] G. I. Webb, J. R. Boughton, and Z. Wang. Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005. 2, 18, 34, 56, 66

[177] G. I. Webb and P. Conilione. Estimating bias and variance from data, 2006. 88

[178] R. White, J. M. Jose, and I. Ruthven. Using top-ranking sentences to facilitate effective information access. *Journal of the American Society for Information Science and Technology*, 56(10):1113–1125, 2005. 163, 164

[179] R. W. White, J. M. Jose, C. J. van Rijsbergen, and I. Ruthven. A simulated study of implicit feedback models. In *Proceedings of the 26th Annual European Conference on Information Retrieval*, pages 311–326, Sunderland, UK, 2004. 178

[180] R. W. White, B. Kules, S. Drucker, and m. c. schraefel. Supporting exploratory search: A special issue of the communications of the acm. *Communications of the ACM*, 49(4):36–39, 2006. 141

[181] R. W. White, I. Ruthven, and J. M. Jose. A study of factors affecting the utility of implicit relevance feedback. In *Proceedings of the 28th SIGIR Conference*, pages 35–42, Salvador, Brazil, 2005. ACM. 146

[182] J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley and Sons, 1990. 10

[183] P. Wilks and M. English. Accurate segmentation of respiration waveforms from infants enabling identification and classification of irregular breathing patterns. *Medical Engineering and Physics, 16(1):19–23*, 1994. 19

[184] S. S. Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *Annals of Mathematical Statistics*, 9:60–62, 1938. 39

[185] I. H. Witten and E. Frank. *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000. 47, 60, 77, 79, 88

[186] D. H. Wolpert. The lack of a priori distinctions between learning algorithms. *Neural Comput.*, 8(7):1341–1390, 1996. 3, 33

[187] S. Wong. Testing implication of probabilistic dependencies. In *Proceedings of the Twelfth Annual Conference on Uncertainty in Artificial Intelligence (UAI–96)*, pages 545–553, Portland, Oregon, 1996. 9

[188] K. S. Woods, C. C. Doss, K. W. Vowyer, J. L. Solka, C. E. Prieve, and W. P. J. Kegelmeyer. Comparative evaluation of pattern recognition techniques for detection of microcalcifications in mammography. *International*

*Journal of Pattern Recognition and Artificial Intelligence, 7(6):1417–1436*, 1993. 19

[189] G. Wright et al. A gene expression-based method to diagnose clinically distinct subgroups of diffuse large B cell lymphoma. *Proceedings of National Academy of Sciences of the USA*, 100(17):9991–6, 2003. xiv, 116, 120, 122, 123, 124, 129, 130, 132, 133, 134

[190] E. P. Xingy, M. I. Jordanyz, and R. M. Karpy. Feature selection for high-dimensional genomic microarray data. In M. Kaufman, editor, *Proceedings of 18th International Conference on Machine Learning*, 2001. 30

[191] C. Yoo, V. Thorsson, and G. F. Cooper. Discovery of causal relationships in a gene-regulation pathway from a mixture of experimental and observational dna microarray data. In *Pacific Symposium on Biocomputing*, pages 498–509, 2002. 106

[192] B. T. Zhang and K. B. Hwang. Bayesian network classifiers for gene expression analysis. *A Practical Approach to Microarray Data Analysis*, pages 150–165, 2003. 107

[193] H. Zhang, L. Jiang, and J. Su. Hidden naive bayes. In *AAAI*, pages 919–924, 2005. 34, 57

[194] H. Zhang, C.-Y. Yu, and B. Singer. Cell and tumor classification using gene expression data: Construction of forests. *Proceedings of the National Academy of Sciences*, 100:4168–4172, 2003. 121, 132

[195] F. Zheng and G. Webb. A comparative study of semi-naive bayes methods in classification learning. In *Proceedings 4th Australasian Data Mining conference (AusDM05)*, pages 141–156, 2005. 9, 11, 15, 16, 17, 18, 33, 34

[196] Z. Zheng, G. I. Webb, and K. M. Ting. Lazy Bayesian rules: a lazy semi-naive Bayesian learning technique competitive to boosting decision trees. In *Proceedings 16th International Conference on Machine Learning*, pages 493–502. Morgan Kaufmann, San Francisco, CA, 1999. 34, 55