University of Kent

Novel Hierarchical Feature Selection Methods for Classification and Their Application to Datasets of Ageing-Related Genes

A THESIS SUBMITTED TO
THE UNIVERSITY OF KENT AT CANTERBURY
IN THE SUBJECT OF COMPUTER SCIENCE
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Cen Wan

August 2015

Contents

List of Figures

List of Tables

Abstract

Hierarchical Feature Selection (HFS) is an under-explored subarea of data mining/machine learning. Unlike conventional (flat) feature selection algorithms, HFS algorithms work by exploiting hierarchical (generalisation-specialisation) relationships between features, in order to try to improve the predictive accuracy of classifiers. The basic idea is to remove hierarchical redundancy between features, where the presence of a feature in an instance implies the presence of all ancestors of that feature in that instance. By using an HFS algorithm to select a feature subset where the hierarchical redundancy among features is eliminated or reduced, and then giving only the selected feature subset to a classification algorithm, it is possible to improve the predictive accuracy of classification algorithms.

In terms of applications, this thesis focuses on datasets of ageing-related genes. This type of dataset is an interesting type of application for data mining methods due to the technical difficulty and ethical issues associated with doing ageing experiments with humans and the strategic importance of research on the biology of ageing - since age is the greatest risk factor for a number of diseases, but is still a not well understood biological process.

This thesis offers contributions mainly to the area of data mining/machine learning, but also to bioinformatics and the biology of ageing, as discussed next. The first and main type of contribution consists of four novel HFS algorithms, namely: select Hierarchical Information Preserving (HIP) features, select Most Relevant (MR) features, the hybrid HIP–MR algorithm, and the Hierarchy-based Redundancy Eliminated Tree Augmented Naïve Bayes (HRE–TAN) algorithm. These algorithms perform lazy learning-based feature selection - i.e. they postpone the learning process to the moment when testing instances are observed and select a specific feature subset for each testing instance. HIP, MR and HIP–MR

select features in a data pre-processing phase, before running a classification algorithm, and they select features that can be used as input by any lazy classification algorithm. In contrast, HRE–TAN is a feature selection process embedded in the construction of a lazy TAN classifier.

The second type of contribution, relevant to the areas of data mining and bioinformatics, consists of two novel algorithms that exploit the pre-defined structure of the Gene Ontology (GO) and the results of a flat or hierarchical feature selection algorithm to create the network topology of a Bayesian Network Augmented Naïve Bayes (BAN) classifier. These are called GO–BAN algorithms.

The proposed HFS algorithms were in general evaluated in combination with lazy versions of three Bayesian network classifiers, namely Naïve Bayes, TAN and GO–BAN - except that HRE–TAN works only with TAN. The experiments involved comparing the predictive accuracy obtained by these classifiers using the features selected by the proposed HFS algorithms with the predictive accuracy obtained by these classifiers using the features selected by flat feature selection algorithms, as well as the accuracy obtained by the classifiers using all original features (without feature selection) as a baseline.

The experiments used a number of ageing-related datasets, where the instances being classified are genes, the predictive features are GO terms describing hierarchical gene functions, and the classes to be predicted indicate whether a gene has a pro-longevity or anti-longevity effect in the lifespan of a model organism (yeast, worm, fly or mouse).

In general, with the exception of the hybrid HIP–MR which did not obtain good results, the other three proposed HFS algorithms (HIP, MR, HRE–TAN) improved the predictive performance of the baseline Bayesian network classifiers - i.e. in general the classifiers obtained higher accuracies when using only the features selected by the HFS algorithm than when using all original features.

Overall, the most successful of the four HFS algorithms was HIP, which outperformed all other (hierarchical or flat) feature selection algorithms when used in combination with each of the Naïve Bayes, TAN and GO–BAN classifiers. The difference of predictive accuracy between HIP and the other feature selection algorithms was almost always statistically significant - except that the difference of accuracy between HIP and MR was not significant with TAN.

Comparing different combinations of a HFS algorithm and a Bayesian network

classifier, HIP+NB and HIP+GO–BAN were both the best combination, with the same average rank across all datasets. They obtained predictive accuracies statistically significantly higher than the accuracies obtained by all other combinations of HFS algorithm and classifier.

The third type of contribution of this thesis is a contribution to the biology of ageing. More precisely, the proposed HIP and MR algorithms were used to produce rankings of GO terms in decreasing order of their usefulness for predicting the pro-longevity or anti-longevity effect of a gene on a model organism; and the top GO terms in these rankings were interpreted with the help of a biologist expert on ageing, leading to potentially relevant patterns about the biology of ageing.

Acknowledgements

Firstly, I would like to sincerely acknowledge my parents and the whole family. They are always beside me, guiding me and supporting me. I sincerely acknowledge my supervisor Prof. Alex A. Freitas, who always inspires me, encourages me, and offered me the enormous support during my PhD.

I would like to acknowledge Dr. João Pedro de Magalhães, who offered great support on this PhD project and co-authored one associated journal paper. Then I would like to acknowledge Daniel Wuttke and Dr. Robi Tacutu, with whom I had valuable discussions about the ageing-related datasets.

In addition, I would like to sincerely appreciate those great people like Dr. Colin Johnson, Prof. Sally Fincher, Prof. Richard Jones, Dr. Fred Barnes and Darren Lissenden, who offered great support on my PhD project.

Finally, I appreciate the School of Computing at University of Kent for offering me a PhD scholarship that sponsored me to complete my PhD project.

Chapter 1

Introduction

Data mining (or machine learning) techniques have attracted considerable attention from both academia and industry, due to their significant contributions to intelligent data analysis. The importance of data mining and its applications is likely to increase even further in the future, given that organisations keep collecting increasingly larger amounts of data and more diverse types of data.

The thesis describes inter-disciplinary research, integrating the areas of data mining and the biology of ageing. Hence, before describing the contributions of this research, we first specify its scope within each of those two areas.

This research addresses the classification task of data mining [37,50,129], where each instance (object being classified) consists of a set of features – sometimes called attributes – and a class variable. The goal of a classification algorithm is to build, from a set of training instances (called the training set), a classification model that predicts the value (also called label) of the class variable for an instance, based on the values of the features for that instance. Note that the classification model is built from the training set, where the algorithm has access to the class label of each instance; but the model is evaluated on a separate set of instances (called the testing set), where the algorithm does not have access to the class label of each instance – those class labels will have to be predicted, as mentioned earlier. After these predictions are computed for all instances in the testing set, the system computes the accuracy of those predictions, by comparing the class label predicted for each testing instance with that instance's true class label. Hence, the testing set is used to measure the predictive performance, or

generalisation ability, of the model built from the training set.

In the context of the classification task, this thesis focuses on the feature selection task. When the number of features is large (like in the datasets used in this research), it is common to apply feature selection methods to the data. These methods aim at selecting, out of all available features in the dataset being mined, a subset of the most relevant and non-redundant features [84,96] for classifying instances in that dataset. There are several motivations for feature selection [84,96], one of the main motivations is to try to improve the predictive performance of classifiers. Another motivation is to accelerate the training time for building the classifiers, since training a classifier with the selected features should be considerably faster than training the classifier with all original features, in general. Yet another motivation is that the selected features may represent a type of knowledge or pattern by themselves, i.e. users may be interested in knowing the most relevant features in their datasets.

Note that feature selection is a hard computational problem, since the number of candidate solutions (feature subsets) grows exponentially with the number of features. More precisely, the number of candidate solution is 2^m - 1, where m is the number of available features in the dataset being mined, and "1" is subtracted in order to take into account that the empty subset of features is not a valid solution for the classification task.

Although there are many types of feature selection methods for classification [47, 84, 96], in general these methods have the limitation that they do not exploit information associated with the hierarchy (generalisation-specialisation relationships) among features, which present in some types of features. As the example shown in Figure 1.1, those features like J, H, D, B, A, C, etc., are hierarchically structured as a Directed Acyclic Graph (DAG), where feature J is the parent of features H and D, and both of them are the parent of feature B, while feature A is the child of features D and C.

This type of hierarchical relationships are relatively common (although usually ignored) in applications. In text mining, for instance, features usually represent the presence or absence of words in a document, and words are involved in generalisation-specialisation relationships [31, 91]; in bioinformatics, which is the type of application this thesis focuses on, the functions of genes or proteins are often described by using a hierarchy of terms, where terms representing more generic

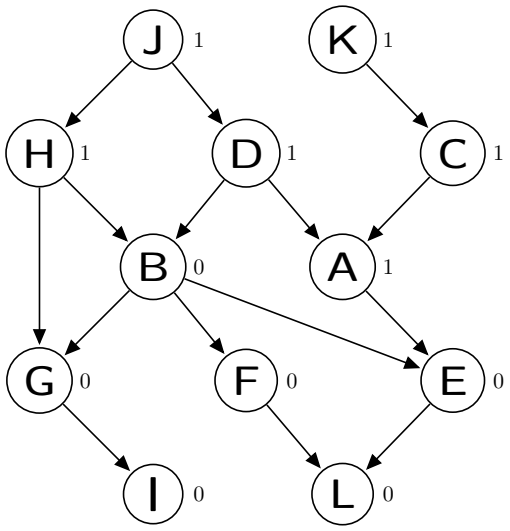


FIGURE 1.1 Example of a Small DAG of Features

functions are ancestors of terms representing more specific functions. As another example of hierarchical features, many datasets in financial or marketing applications (where instances represent customers) have the address of the customer as a feature. This feature can be specified at several hierarchical levels, varying from the most detailed level (e.g. the full post code) to more generic levels (e.g. the first two or first three digits of the post code).

From another perspective, hierarchies of features can also be produced by using hierarchical clustering algorithms [129] to cluster features, rather than to cluster instances, based on a measure of similarity between features. The basic idea is that each object to be clustered would be a feature, and the similarity between any two features would be given by a measure of how similar the values of those features are across all instances. For instance, consider a dataset where each instance represents an email, and each binary feature represents the presence or absence of a word. Two features (words) can be considered similar to the extent that they occur (or don't occur) in the same sets of emails. Then, a hierarchical clustering algorithm can be used to produce a hierarchy of features, where each leaf cluster will consist of a single word, and higher-level clusters will consist of a list of words connected by an "or" logical operator. For example, if words "money" and "buy" were merged into a cluster by the hierarchical clustering algorithm, when mapping the original features to the hierarchical features created by the clustering algorithm, an email with word "money" but without the word "buy" would be considered to have value "yes" for feature "money", value "no" for feature "buy",

and value "yes" for feature "money or buy". Note that in this example the "or" operator was used (as opposed to the "and" operator) in order to make sure the feature hierarchy is a "IS-A" hierarchy; i.e. if an email has value "yes" for a feature, it will necessarily have value "yes" for all ancestors of that feature in the hierarchy.

Intuitively, in datasets where such hierarchical relations among features exist, ignoring such relationships seems a sub-optimal approach; i.e. these hierarchical relationships represent additional information about the features that could be exploited to improve the predictive performance associated with feature selection methods – i.e. the ability of these methods to select features that maximise the predictive accuracy to be obtained by classification algorithms using the selected features. This is the basic idea behind the hierarchical feature selection methods proposed in this thesis.

The proposed hierarchical feature selection methods perform "lazy learning", in the sense that they postpone the feature selection process to the moment when testing instances are observed, rather than in the training phase of conventional learning methods (which perform "eager learning"). The proposed methods are evaluated together with lazy learning versions of Bayesian network classifiers (although other types of lazy learning classifiers could be used in feature research).

In terms of applications of the proposed hierarchical feature selection methods, this thesis focuses on analysing biological data about ageing-related genes [27,30,35,54,82,122–124]. The causes and mechanisms of the biological process of ageing are a mystery that has puzzled humans for a long time. Biological research has, however, revealed some factors that seem associated with the ageing process.

For instance, caloric restriction – which consists of taking a reduced amount of calories without undergoing malnutrition – extends the longevity of many species [88]. In addition, research has identified that several biological pathways seem to regulate the process of ageing (at least in model organisms), such as the well-known insulin/insulin-like growth factor (IGF-1) signalling pathway [68]. It is also known that mutations in some DNA repair genes lead to accelerated ageing syndromes [34]. Despite such findings, ageing is a highly complex biological process which is still poorly understood, and much more research is needed in this area.

Unfortunately, conducting ageing experiments in humans is very difficult, due to the complexity of the human genome, the long lifespan of humans, and ethical issues associated with experiments with human. Therefore, research on the

biology of ageing is usually done with model organisms like yeast, worms, flies or mice, which can be observed in an acceptable time and have considerably simpler genomes. In addition, with the growing amount of ageing-related data on model organisms available on the web, in particular related to the genetics of ageing, it is timely to apply data mining methods to that data [123], in order to try to discover patterns that may assist ageing research.

More precisely, in this work, the instances being classified are genes from four major model organisms, namely: *C. elegans*, *S. cerevisiae*, *D. melanogaster* and *M. musculus*. Each gene has to be classified into one of two classes: pro-longevity or anti-longevity, based on the values of features indicating whether or not the gene is associated with each of a number of Gene Ontology (GO) terms, where each term refers to a type of biological process, molecular function or cellular component. Pro-longevity genes are those whose decreased expression (due to knockout, mutations or RNA interference) reduces lifespan and/or whose overexpression extends lifespan; accordingly, anti-longevity genes are those whose decreased expression extends lifespan and/or whose overexpression decreases it [111].

We adopt GO terms as features to predict a gene's effect on longevity because of the widespread use of the GO in gene and protein function prediction and the fact that GO terms were explicitly designed to be valid across different types of organisms [112]. GO terms are organised into a hierarchical structure where, for each GO term t, its ancestors in the hierarchy denote more general terms (i.e. more general biological processes, molecular function or cellular component) and its descendants denote more specialised terms than t. It is important to consider the hierarchical relationships among GO terms when performing feature selection, because such relationships encode information about redundancy among GO terms. In particular, if a given gene g is associated with a given GO term t, this logically implies that is also associated with all ancestors of t in the GO hierarchy. This kind of redundancy can have a substantially negative effect on the predictive accuracy of Bayesian network classification algorithms, such as Naïve Bayes [129]. This issue will be discussed in detail later.

1.1 An Overview of Original Contributions

This thesis makes original contributions in terms of proposing and empirically evaluating four hierarchical feature selection methods, including three filter methods (which run in a data pre-processing phase, independent of the classifier), described in Chapter 4; and one embedded method (i.e. a method that performs the feature selection process as part of the process of building the classifier), described in Chapter 5. In addition to these hierarchical feature selection methods, two algorithms for constructing the network topology of a Bayesian Network Augmented Naïve Bayes classifier are also proposed and empirically evaluated in Chapter 6. Both these methods are based on the features selected by conventional flat or the new hierarchical feature selection methods. Note that these contributions, which are the main contributions of this thesis, are contributions to the area of machine learning/data mining.

As another type of contributions, which are contributions to the area of the biology of ageing, we have created new datasets of ageing-related genes with hierarchical features, in order to evaluate the proposed hierarchical feature selection methods. In addition, these methods were applied to the created datasets, and the results were used to produce rankings of biological features.

1.2 Structure of This Thesis

This thesis is structured into 7 chapters, including the current *Introduction Chapter*. A brief description of the remaining chapters is presented next.

- **Chapter 2 - Background on Data Mining**
  This chapter presents a review of data mining concepts and methods relevant for this research, especially focusing on the classification task. Conventional types of Bayesian network classification algorithms, e.g. Naïve Bayes and some Semi-naïve Bayes classifiers will be discussed. Moreover, feature selection methods for classification will also be discussed in detail.

- **Chapter 3 - Background on Biology of Ageing and Bioinformatics**
  This chapter presents a brief review about molecular biology, the biology of

ageing and bioinformatics, especially focusing on the task of gene/protein function prediction. Then, related works about ageing-related gene/protein function prediction using machine learning/data mining methods as well as work on classification methods applied to the biology of ageing will be reviewed.

- **Chapter 4 - Lazy Hierarchical Feature Selection Methods with Naïve Bayes**

  This chapter presents a detailed description of three proposed filter hierarchical feature selection methods, followed by the empirical evaluation of their predictive performance when working with the Naïve Bayes classifier, in a number of ageing-related datasets. This chapter also presents the methods used to create the ageing-related datasets that were used in our experiments. In addition, this chapter also reports a ranking of ageing-related GO terms, based on the results of one of the best performing hierarchical feature selection methods.

- **Chapter 5 - Lazy Hierarchical Feature Selection Methods with Tree Augmented Naïve Bayes (TAN)**

  This chapter presents a detailed description of one proposed embedded hierarchical feature selection method based on the Tree Augmented Naïve Bayes (TAN) classifier, followed by the empirical evaluation of its predictive performance by comparing it with other feature selection methods (including the filter hierarchical feature selection methods proposed in Chapter 4), when working with the Tree Augmented Naïve Bayes (TAN) classifier. This chapter also reports a ranking of ageing-related GO terms, based on one of the best performing hierarchical feature selection methods combined with the TAN classifier.

- **Chapter 6 - Lazy Hierarchical Feature Selection Methods with Bayesian Network Augmented Naïve Bayes (BAN)**

  This chapter presents a detailed description of two algorithms proposed for constructing the network topology of a Gene Ontology-based Bayesian Network Augmented Naïve Bayes (GO–BAN), based on the features selected by either flat or hierarchical feature selection methods. This chapter also

conducts an empirical evaluation of both proposed algorithms. In addition, this chapter includes a comparison between the best performing hierarchical feature selection methods when working with different Bayesian network classifiers, i.e. Naïve Bayes, Tree Augmented Naïve Bayes and Gene Ontology-based Bayesian Network Augmented Naïve Bayes.

- **Chapter 7 - Conclusions and Future Work**
  This chapter concludes the thesis by summarising its contributions to the area of machine learning/data mining (primary contribution) and the area of biology/bioinformatics of ageing research (secondary contribution). In addition, further research directions are suggested.

1.3 List of Publications

The publications derived from this thesis consist of one journal paper, two conference papers and one abstract. In addition, one journal paper is in preparation. The detailed information about these papers is listed below.

***Peer-Reviewed Journal Paper*:**

- **C. Wan**, A. A. Freitas, and J. P. de Magalhães, "Predicting the Pro-longevity or Anti-longevity Effect of Model Organism Genes With New Hierarchical Feature Selection Methods", ***IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)***, 12(2), pp. 262–275, Mar.–Apr., 2015. DOI: 10.1109/TCBB.2014.2355218.
  Note: This paper is a major extension of the IEEE BIBM conference paper, whose details are mentioned later.

***Journal Paper in Preparation*:**

- **C. Wan** and A. A. Freitas, "An Empirical Evaluation of Hierarchical Feature Selection Methods in Datasets of Ageing-related Genes".

Peer-Reviewed Conference Papers:

- **C. Wan** and A. A. Freitas, "Prediction of the pro-longevity or anti-longevity effect of *Caenorhabditis Elegans* genes based on Bayesian classification methods", in Proceedings of ***IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2013)***, Shanghai, China, Dec., 2013, pp. 373–380. (**Acceptance rate: 19.6%, 60/306**)

- **C. Wan** and A. A. Freitas, "Two Methods for Constructing a Gene Ontology-based Feature Network for a Bayesian Network Classifier and Applications to Datasets of Aging-related Genes", in Proceedings of ***the 6th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM–BCB 2015)***, Atlanta, USA, Sept., 2015, pp. 27–36. (**Acceptance rate: 34.0%, 48/141**)

Published Abstract:

- **C. Wan** and A. A. Freitas. Gene Ontology Hierarchy-based Feature Selection. ***Features and Structures 2014 (FEAST 2014) Workshop attached to the 22nd International Conference on Pattern Recognition (ICPR 2014)***, Stockholm, Sweden, Aug., 2014. (Abstract [125]; Poster and Oral Presentation)

Chapter 2

Background on Data Mining

2.1 Knowledge Discovery in Databases (KDD)

Due to the rapid growth of data from real world applications, it is timely to adopt *Knowledge Discovery in Databases (KDD)* methods to extract knowledge or valuable information from data. Indeed, KDD has already been successfully adopted in real world applications, both in science and in business.

KDD is a field of inter-disciplinary research across machine learning, statistics, databases, etc [37,50,129]. Broadly speaking, the KDD process can be divided into four phases. The first phase is selecting raw data from original databases according to a specific knowledge discovery task, e.g. classification, regression or clustering. Then the selected raw data will be input to the phase of data pre-processing (the second phase), which aims at processing the data into a form that could be efficiently used by the type of algorithm(s) to be applied in the data mining phase - such algorithms are dependent on the chosen type of knowledge discovery task. The data pre-processing phase includes data cleaning, data normalisation, feature selection and feature extraction, etc. The third phase is data mining, where a model will be built by running learning algorithms on the pre-processed data. In this work, we address the classification task, where the learning (classification) algorithm builds a classification model or classifier as will be explained later. The final phase is extracting the knowledge from the built classifier or model. Among those four phases of KDD, the focus of this research is on the data pre-processing

phase, in particular the feature selection task, where the goal is to remove the redundant or irrelevant features in order to improve the predictive performance of classifiers. The feature selection task will be reviewed later in this chapter.

2.2 Data Mining Tasks and Paradigms

Data Mining tasks are types of problems to be solved by a machine learning or data mining algorithm. The main types of data mining tasks can be categorized as *classification*, *regression* and *clustering*. The former two tasks (*classification* and *regression*) are also grouped as the *supervised learning* paradigm, whereas the latter one (*clustering*) is categorised as *unsupervised learning*.

Supervised learning consists of learning a function from labeled training data [93]. The supervised learning process consists of two phases, i.e. the training phase and the testing phase. Accordingly, in the supervised learning process, the original dataset is divided into training and testing datasets. In the training phase, only the training dataset will be used for inferring the specific function by learning a specific model, which will be evaluated by using the testing dataset in the testing phase.

Unlike supervised learning, unsupervised learning is usually defined as a process of learning particular patterns from unlabelled data. In unsupervised learning, there is no distinction between training and testing datasets, and all available data are used to build the model. The usual application of unsupervised learning is to find groups (or clusters)/patterns of similar instances, constituting a clustering problem.

2.2.1 Classification

The classification task is possibly the mostly studied task in data mining. It consists of building a classification model or classifier to predict the class label (a nominal or categorical value) of an instance by using the values of the features (predictor attributes) of that instance [37, 50]. Actually, the essence of the classification process is exploiting correlations between features and the class labels of instances in order to find the border between class labels in the data space - a

space where the position of an instance is determined by the values of the features in that instance. The classification border is exemplified in Figure 2.1, in the context of a problem with just two class labels, where the found classification border (a black dashed line) distinguishes the instances labelled as square or circle.

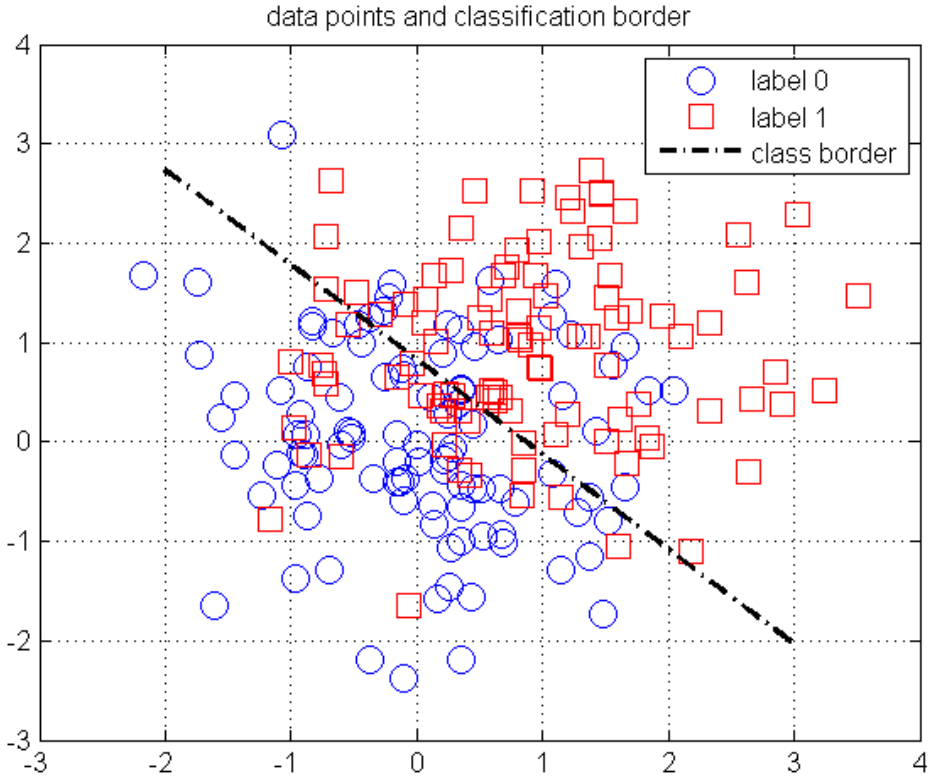


FIGURE 2.1 Example of Data Classifiertion into Two Categories [89]

Many types of classification algorithms have been proposed, such as Bayesian network classifiers, Decision Trees, Support Vector Machines (SVM), Artificial Neural Networks (ANN), etc. From the perspective of interpretability of the classifier, those classifiers can be categorised into two groups, i.e. "white box" and "black box" classifiers. The "white box" classifiers, e.g. Bayesian network classifiers and Decision Trees, have better interpretability than the latter ones, e.g. Support Vector Machine (SVM) and Artificial Neural Networks (ANN) [38]. In this thesis, we focus on Bayesian network classifiers [41, 126, 127, 141, 142] (more precisely, Naïve Bayes and Semi-naïve Bayes classifiers), due to their good potential for interpretability; in addition to their ability to cope with uncertainty in data – a common problem in bioinformatics [44].

2.2.2 Regression

Regression analysis is a traditional statistical task with the theme of discovering the association between predictive variables (features) and the target (response) variable. As it is usually used for prediction, regression analysis can also be considered a type of supervised learning task from the perspective of machine learning and data mining.

Overall, a regression method is capable of predicting the numeric (real-valued) value of the target variable of an instance - unlike classification methods, which predict nominal (categorical) values, as mentioned earlier. A typical example of a conventional linear regression model for a dataset with just one feature x is shown as Equation 2.1,

$$y_i = \beta_0 + \beta_1 x_i + \xi_i \tag{2.1}$$

where x_i denotes the value of the feature x for the *i-th* instance, β_i denotes the corresponding weight, and ξ_i denotes the error. The most appropriate values of the weights in Equation 2.1 can be found using mathematical methods, such as the well-known Linear Least Square [78,86,110]. Then the predicted output value y_i is computed based on the values of the input feature with its corresponding weight. As shown in the simple example of Figure 2.2, the small distances between the line and the data points indicates that Equation 2.1 fits well the data. Regression analysis has been well studied in the statistics area and widely applied in different domains.

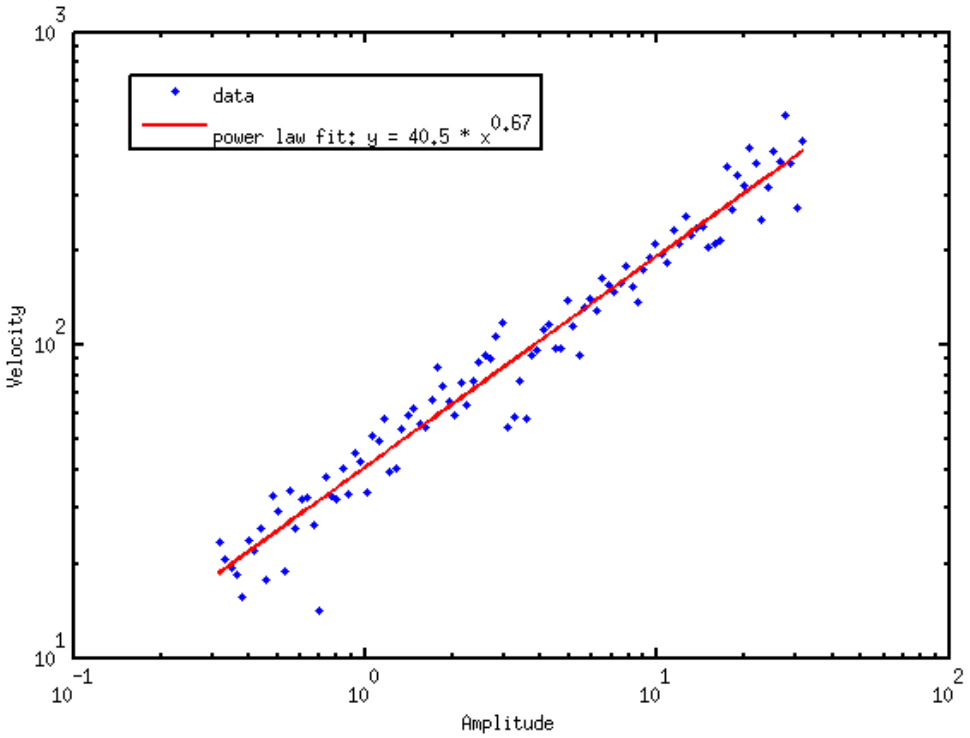


FIGURE 2.2 Example of Regression for Data [56]

2.2.3 Clustering

The clustering task mainly aims at finding patterns in the data by grouping similar instances into clusters (or groups). The instances within the same cluster are more similar with each other, but simultaneously more dissimilar with the instances in other clusters. An example of clustering is shown in Figure 2.3, where the left graph represents the situation before clustering, where all data are unlabelled (in blue), and the right graph represents the situation where all data are clustered into three different groups, i.e. one group of data in blue, one group of data in red, and one group of data in green.

Clustering has been widely studied in the area of statistical data analysis, and applied on different domains, like information retrieval, bioinformatics, etc. Examples of well-known, classical clustering methods are k-means [51] and k-medoids [61].

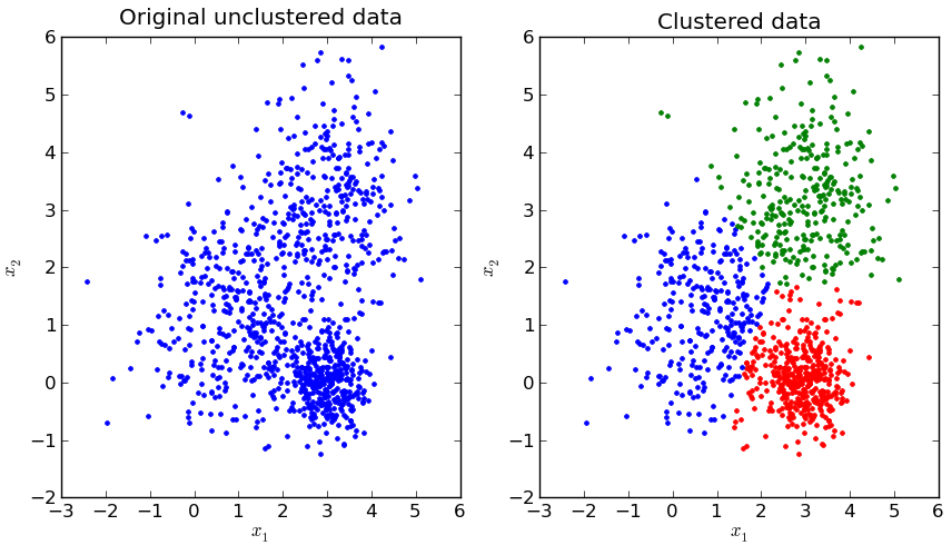


FIGURE 2.3 Example of Data Clustered into Three Groups [99]

2.2.4 Eager and Lazy Learning Paradigms

Data mining or machine learning methods can be categorised into two general paradigms, depending on when the learning process is performed, namely: eager learning and lazy learning. An eager learning method performs the learning process during the training phase, i.e. learning the classifier (or classification model) using the whole training dataset before any testing instance is observed. Then the classifier is used to classify all testing instances. This is in contrast to the lazy learning approach, where the learning process is performed after observing the feature values for each individual testing instance in the testing phase. That is, a lazy learning-based classification algorithm builds a specific classification model for each individual testing instance to be classified [6, 96].

In the context of feature selection, which is the research theme of this thesis and will be discussed in later sections, lazy learning-based methods select a specific set of features for each individual testing instance, whilst eager learning-based methods select a single set of features for all testing instances.

2.3 The Naïve Bayes (NB) Classifier

The Naïve Bayes classifier [37,50,92,95,129] is a type of Bayesian network classifier that assumes that all features are independent from each other given the class attribute. An example of this classifier's network topology is shown in Figure 2.4, where each feature x_i $(i = 1, 2, ..., 5)$ only depends on the class attribute. In the figure, this is indicated by an edge pointing from the class node to each of the feature nodes. As shown in Equation 2.2,

$$P(y \mid x_1, x_2, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid y) \tag{2.2}$$

where $\propto$ is the mathematical symbol for proportionality and n is the number of features; the estimation of the probability of a class attribute value y given all predictor features' values x_i of one instance can be obtained by calculating the product of the individual probability of each feature value given a class attribute value and the prior probability of that class attribute value. Naïve Bayes (NB) has been shown to have relatively powerful predictive performance, compared with other Bayesian network classifiers [41], even thought it pays the price of losing the dependencies between features.

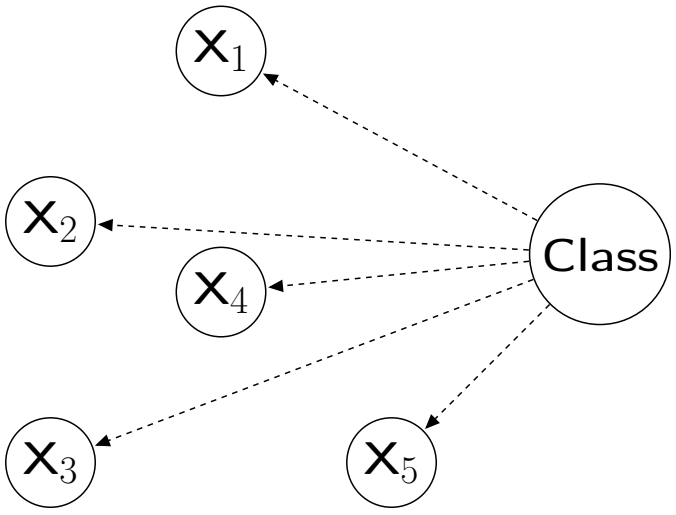


FIGURE 2.4 An Example Naïve Bayes Network Topology

2.4 Semi-naïve Bayes Classifiers

The Naïve Bayes classifier is very popular and has been applied on many domains due to its advantages of simplicity and short learning time, compared with other Bayesian classifiers. However, the assumption of conditional independence between features is usually violated in practice. Therefore, many extensions of Naïve Bayes focus on approaches to relax the assumption of conditional independence [41, 73, 141]. This sort of classifier is called Semi-naïve Bayes classifier.

Both the Naïve Bayes classifier and Semi-naïve Bayes classifiers use estimation of the prior probability of the class and the conditional probability of the features given the class to obtain the posterior probability of the class given the features, as shown in the Equation 2.3 (i.e. the Bayes' formula), where y denotes a class and x denotes the set of features, i.e. $\{x_1, x_2, ..., x_n\}$. However, different Semi-naïve Bayes classifiers use different approaches to estimate the term $P(x \mid y)$, as discussed in the next subsections.

$$P(y \mid x) = \frac{P(x \mid y)P(y)}{P(x)} \tag{2.3}$$

2.4.1 Tree Augmented Naïve Bayes (TAN) and SuperParent Tree Augmented Naïve Bayes (SP–TAN)

TAN constructs a network in the form of a tree, where each feature node is allowed to have at most one parent feature node in addition to the class node (which is a parent of all feature nodes), as shown in Figure 2.5, where each feature except the root feature X_4 has only one non-class parent feature. TAN computes the posterior probability of a class y using Equation 2.4,

$$P(y \mid x_1, x_2, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i \mid Par(x_i), y) \tag{2.4}$$

where the number of non-class parent features for each feature x_i (i.e. $Par(x_i)$), except the root feature, equals to "1". Hence, it represents a limited degree of dependencies among features.

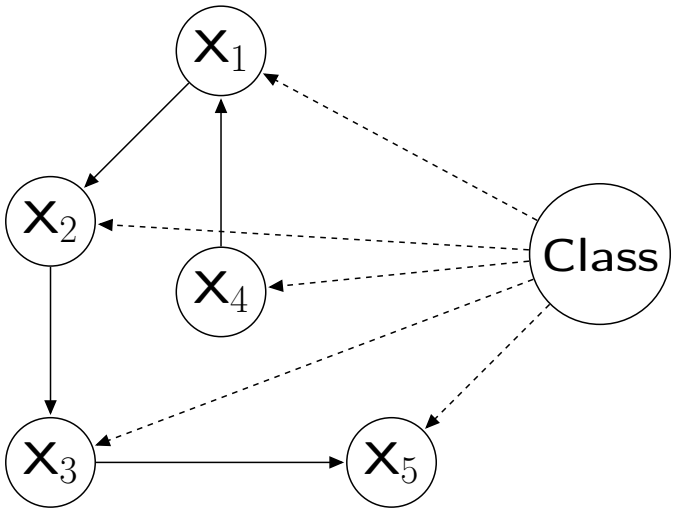


FIGURE 2.5 An Example of TAN's Network Topology

In essence, the original TAN classifier firstly produces a rank of feature pairs according to the conditional mutual information between the pair of features given the class attribute. Then the Maximum Spanning Tree is built based on the rank. Next, the algorithm randomly chooses a root feature and then sets all directions of edges to other features from it. Finally, the constructed tree is used for classification.

The concept of conditional mutual information proposed for building TAN classifiers is an extension of mutual information. The formula of conditional mutual information is shown as Equation 2.5,

$$I_p(X_i; X_j \mid Y) = \sum_{x_i, x_j, y} P(x_i, x_j, y) log \frac{P(x_i, x_j \mid y)}{P(x_i \mid y) P(x_j \mid y)} \tag{2.5}$$

where X_i and X_j are predictor features, Y is the class attribute, x_i, x_j, y are the values of the corresponding features and the class attribute, $P(x_i, x_j, y)$ denotes the joint probability of x_i, x_j, y; $P(x_i, x_j \mid y)$ denotes the joint probability of feature values x_i and x_j given class value y; and $P(x_i \mid y)$ denotes the conditional probability of feature value x_i given class value y. Each pair of features "x_i, x_j" is taken into account as a group, then the mutual information for each pair of features given the class attribute is computed [41].

As a variant of TAN, SuperParent-TAN (SP–TAN) adopts the wrapper approach to build the feature tree. More precisely, it tentatively makes each feature

node as the *SuperParent* in turn. The *SuperParent* is a node that has arcs to every *orphan* node, i.e. every node that currently has no feature parent. Then, the node that mostly improves the predictive accuracy by leave-one-out cross validation will be selected as the *SuperParent* A_{sp}. After selecting the unique *SuperParent* feature, the selection of its *favorite orphan* is conducted. The *favorite orphan* is the feature which mostly improves the predictive accuracy, if it is connected with the *SuperParent*. Then an arc will be connected from A_{sp} to its *favorite orphan*. The process above will be repeated until there is no improvement on accuracy or the number of remaining *orphans* equals to one [69].

In terms of the type of classification model finally built, the original TAN randomly selects a root node of the Maximum Spanning Tree, whereas SP–TAN selects a *SuperParent* node as the root by taking into account the predictive performance of the feature tree. In the topology of the network built by TAN, the number of arcs equals to n - 1 (n denotes the number of nodes), whereas the number of arcs made by SP–TAN might be fewer. According to the experimental results reported in [69], SP–TAN outperforms TAN in most cases for the datasets adopted in the experiments.

2.4.2 Bayesian Network Augmented Naïve Bayes (BAN)

The BAN classifier is a more complicated type of Semi-naïve Bayes classifier, which (unlike NB and TAN) can represent more complicated dependencies between features [23, 41]. More precisely, in a BAN, in Equation 2.4, the number of parent feature node(s) for each node x_i (i.e. $Par(x_i)$) is allowed to be more than one. An example of this classifier's network topology is shown in Figure 2.6, where each feature x_i has the class attribute as a parent, indicated by the dashed lines; and possibly other non-class parent feature(s), as indicated by the solid lines. Node X_4 has two non-class parent nodes X_1 and X_5, while node X_3 also has two non-class parent nodes X_2 and X_4.

There exist several approaches for constructing a BAN classifier from data that have been shown to be relatively efficient to use, particularly when the number of feature parents of a node is limited to a small integer number (a user-specified parameter). However, in general, learning a BAN classifier tends to be much more time consuming than learning a NB or TAN classifier, mainly due to the large

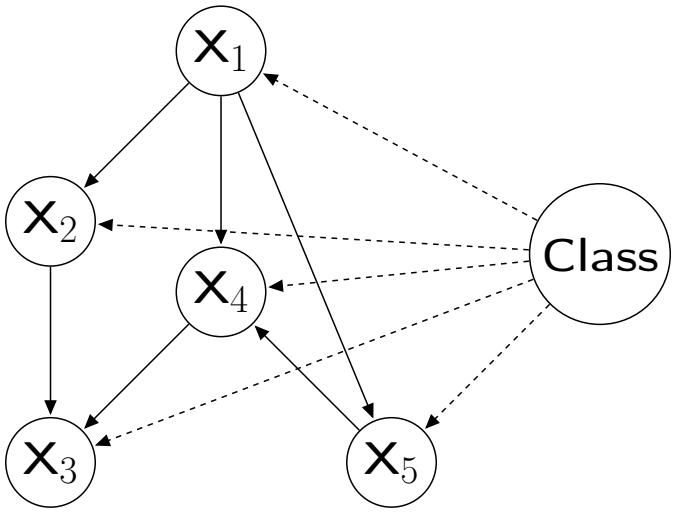


FIGURE 2.6 An Example of BAN's Network Topology

time taken to search for a good BAN network topology.

Fortunately, in the context of the bioinformatics data used in this project, there are strong dependency relationships between features, which have been already defined by expert biologists in the form of a feature graph, containing hierarchical relationships among features that are represented as directed edges in the feature graph (as will be explained in detail later). Such hierarchical relationships provide a sophisticated representation of biological knowledge that can be directly exploited by a BAN classifier. Hence, we will use the pre-defined hierarchical relationships retained in the data as the topology of the BAN classifier network, rather than learning the BAN network topology from the data, as will be discussed in Chapter 6.

2.4.3 Average One-Dependence Estimators (AODE)

The Average One-Dependence Estimators (AODE) method [127] infers the class of a new instance by calculating the average posterior class probability over all possible one-dependence classifiers. An one-dependence classifier consists of merely one feature as the parent for all other features. Each feature is treated as the parent for all other features in turn. For example, in Figure 2.7, five types of AODE's network topology represent the cases where each of features X_1, X_2, ..., X_5 is the parent feature in turn. In this figure, the dependencies between a parent feature and its child features are shown in solid lines, while the dashed lines denote the

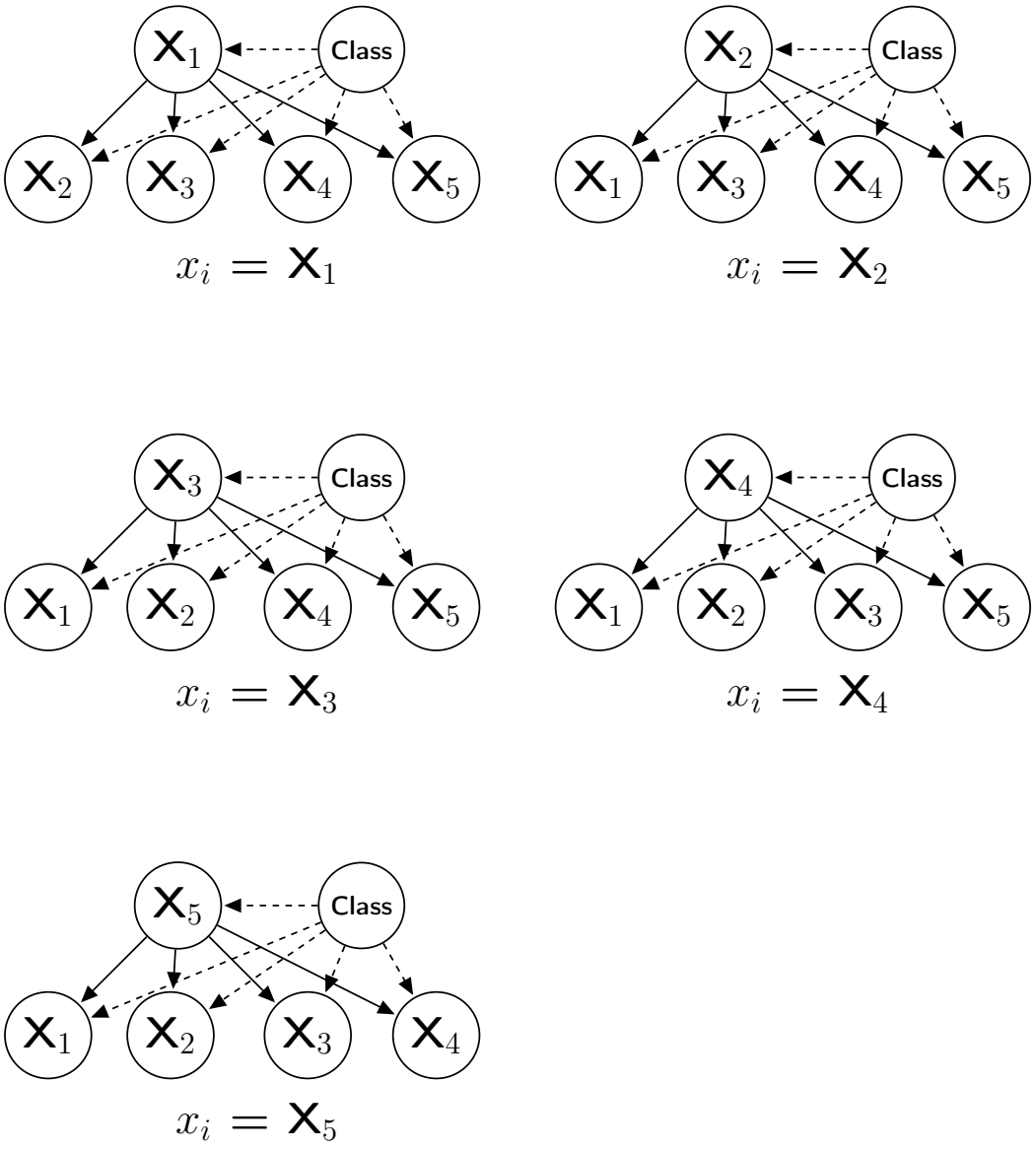


FIGURE 2.7 An Example of AODE's Network Topology

dependencies between the class attribute and all features.

In order to avoid the inaccurate estimation of probabilities caused by few instances, the minimal number of instances that have each value of the parent feature was set to 30, due to concerns on statistical significance. AODE computes the posterior probability of class value y given the values of the set of features x as shown in Equation 2.6,

$$P(y \mid x) \propto \frac{\sum_{i \in N \bigwedge F(x_i) \geq m} P(y, x_i) \prod_{j \in N, j \neq i} P(x_j \mid y, x_i)}{|i : \{i \in N \bigwedge F(x_i) \geq m\}|} \tag{2.6}$$

where x_i denotes each possible parent feature for all other features, x_j denotes one of the features in the set of features except the parent feature x_i, $F(x_i)$ denotes the number of instances associated with different values of the parent feature x_i, N denotes the set of feature indices and m is a user-defined parameter – set to 30 in the original work proposing AODE, as mentioned earlier.

In terms of alleviating the problem of the feature independence assumption for Naïve Bayes, the AODE algorithm has the advantages of simplicity and theoretical foundation. But it has the disadvantage that the model's interpretability is hindered by the fact that the final model actually consists of a large number of one-dependency models (one such model for each predictor feature used as parent for all other features).

2.4.4 Naïve Bayes Tree (NBTree)

The NBTree classifier [72] is a hybrid classifier combining Naïve Bayes and Decision Tree classifiers. It follows the idea of recursive partitioning of a dataset according to the values of features selected to discriminate among the classes, as performed by Decision Tree algorithms [37]. An important difference between NBTree and conventional Decision Tree algorithms is the evaluation function used for selecting features. NBTree uses the utility (rather than the entropy) of individual features as the criterion for selecting the splitting feature. The utility of a feature is measured by the predictive accuracy associated with individual tree nodes by using Naïve Bayes, where the predictive accuracy is estimated through 5-fold cross validation. In NBTree, for each leaf in the tree, the set of features can be divided into two feature subsets, namely the set of splitting features occurring in the path from the root to that leaf, and the remaining set of features (i.e. features not occurring in that path). The estimation of the posterior probability of the class value y given the set of values of the remaining features x_i and the set of values of the splitting features x' for a given leaf is given by Equation 2.7,

$$P(y \mid x, x') \propto P(y, x') \prod_{i \in l} P(x_i \mid y, x') \tag{2.7}$$

where x' is the set of values of the set of splitting features in the path from the root to the current leaf, and l is the set of indices for the remaining features [141]. The utility of each split for an individual feature equals to the weighted sum of the utility of the new leaf nodes created by that split. For the sake of avoiding the over-fitting problem caused by splitting nodes with few instances, the process of recursively splitting the data terminates if the error reduction is below 5% or the number of instances in the current node to be split is less than 30.

According to the experimental results reported in [72], NBTree obtains high predictive accuracy in many cases, but its running time is not competitive against Naïve Bayes. In addition, the interpretability is a merit of NBTree, which is similar to an advantage of Decision Tree classifiers [38].

2.4.5 The Lazy Bayesian Rules (LBR) Algorithm

The Lazy Bayesian Rules (LBR) algorithm [143] follows the lazy learning approach, i.e. it builds a local Naïve Bayes classifier for each testing instance, rather than for the whole training dataset. A rule has the form: **IF**$(antecedent)$, **THEN**$(Class)$; where the Class in the rule's consequent (**THEN** part) is predicted for instances satisfying the rule's antecedent (**IF** part). The antecedent of a Bayesian rule is composed by a set of feature-value pairs with the form "$feature = value$". The utility of adding each feature-value pair into the antecedent is evaluated by leave-one-out cross validation and the best pair will be added into the antecedent if its associated classification error is lower than the error obtained by the existing local Naïve Bayes classifier created from the training dataset. This process terminates if there is no significant improvement on predictive performance. The inference formula used by LBR is shown as Equation 2.8,

$$P(y \mid x, q) \propto P(y, q) \prod_{i \in s} P(x_i \mid y, q) \tag{2.8}$$

where y denotes the class attribute value, q denotes the set of features' values in the rule's antecedent and s represents the set of indices of the remaining features.

LBR's criterion for stopping rule growing can naturally avoid the over-fitting problem by avoiding including in a rule antecedent an infrequent feature value, due

to its "lazy" learning approach. However, LBR uses cross validation to measure the predictive accuracy associated with each feature-value pair to be added to a rule, so it has a high processing time for growing the antecedent.

2.5 Conventional, "Flat" Feature Selection

Feature selection is a type of data pre-processing task that consists of removing irrelevant and redundant features in order to improve the predictive performance of classifiers. The role of feature selection methods in the classification process is illustrated by the flow-chart shown in Figure 2.8, where the dataset with the full set of features is input to the feature selection method, which will select a subset of features to be used for building the classifier. Then the built classifier will be evaluated, by measuring its predictive accuracy. Irrelevant features can be defined as features which are not correlated with the class variable, and so removing such features will not be harmful for the predictive performance. Redundant features can be defined as those features which are strongly correlated with other features, so that removing those redundant features should also not be harmful for the predictive performance.

Generally, feature selection methods can be categorised into three groups, i.e. wrapper approaches, filter approaches and embedded approaches, as discussed next.

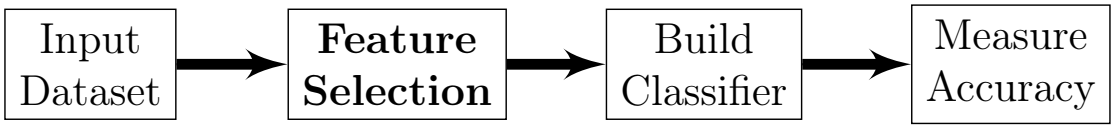


FIGURE 2.8 Flow-Chart of the Classification Process Including Feature Selection in a Pre-Processing Phase

2.5.1 The Wrapper Approach

The wrapper feature selection approach decides which features should be selected from the original full set of features based on the predictive performance of the classifier with different candidate feature subsets. In the wrapper approach, the training dataset is divided into a "building" (or "learning") set and a validation set. As summarised in graphical form in Figure 2.9, the best subset of features to be selected is decided by iteratively getting a candidate feature subset, building the classifier from the learning set, using only the candidate feature subset, and measuring accuracy in the validation set. The boolean function "End?" will check whether the selected subset of features satisfies the expected improvement on predictive performance. If not so, the re-selection of a candidate feature subset will be conducted again, otherwise, the stage of feature selection will terminate, and the best subset of features will be used for building the classifier, which is finally evaluated on the testing dataset.

The wrapper approach selects features that tend to be tailored to the classification algorithm, since the feature selection process was guided by the algorithm's accuracy. However, the wrapper approach has relatively higher time complexity than the filter and embedded approaches, since in the wrapper approach the classification algorithm has to be run many times.

One feature selection method following the wrapper approach is Backward Sequential Elimination (BSE). It starts with the full set of features, then iteratively uses leave-one-out cross validation to detect whether removing a certain feature, whose elimination will most reduce the training error on the validation set, will improve predictive accuracy. It repeats this process until the improvement in accuracy ends [141].

The opposite approach, named Forward Sequential Selection (FSS), starts with the empty set of features and then iteratively adds the feature that mostly improves accuracy on the validation dataset to the set of selected features. This iterative process is repeated until the predictive accuracy starts to decrease [76]. Both wrapper feature selection methods just discussed have a very high processing time because they perform many iterations and each iteration involves measuring predictive accuracy on the validation dataset by running a classification algorithm.

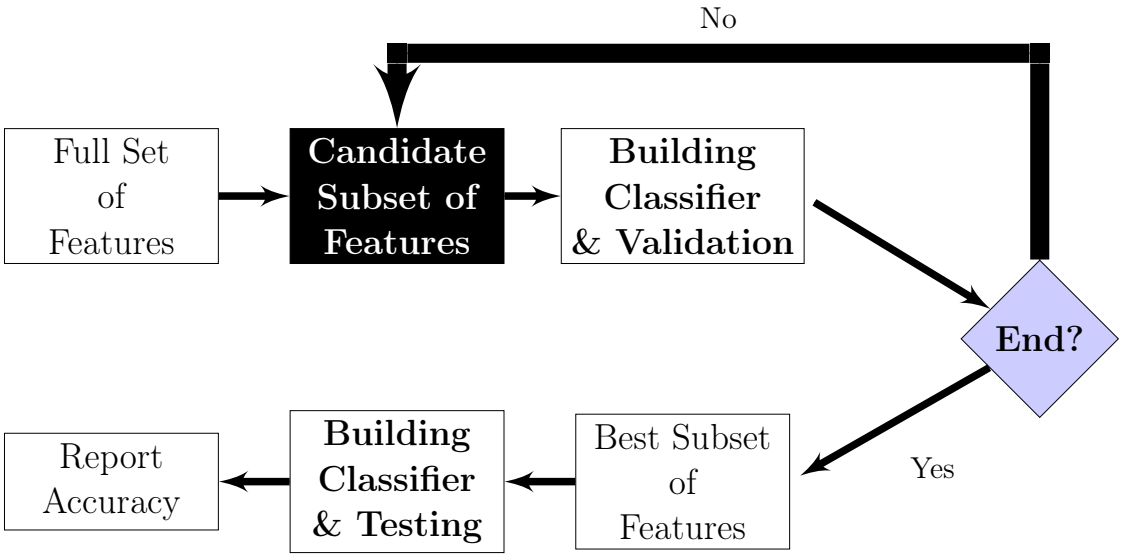


FIGURE 2.9 Flow-Chart of the Wrapper Feature Selection Approach - Adapted from [84]

2.5.2 The Filter Approach

Unlike the wrapper approach, the filter approach conducts the feature selection process by evaluating the quality of a feature or feature subset using a quality measure that is independent from the classification algorithm that will be applied to the selected features. As shown in the flow-chart in Figure 2.10, the subset of features is chosen from the original full set of features according to a certain selection criterion (or feature relevance measure). The selected feature subset is then input into the classification algorithm, the classifier is built and then the predictive accuracy is measured on the testing set and reported to the user. Note that the classifier is built and evaluated only once at the end of the process, rather than being iteratively built and evaluated in a loop, like in the wrapper approach (Figure 2.9). This means the filter approach is much faster than the wrapper approach in general. In this thesis, we propose three filter feature selection methods, which will be described in detail in Chapter 4.

Filter feature selection methods can be mainly categorised into two groups. The first group focuses on measuring the quality (relevance) of each individual feature without taking into account the interaction with other features. Basically,

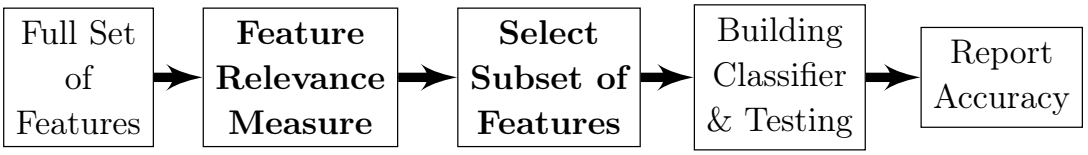


FIGURE 2.10 Flow-Chart of the Filter Feature Selection Approach - Adapted from [84]

the relevance of each feature will be evaluated by a certain criterion, such as the mutual information with the class variable, the information gain [131], etc. Then all features will be ranked in descending order according to the corresponding relevance measure. Only the top-k features will be selected for the classification stage, where k is a user-defined parameter. This type of methods is simple, but it ignores the interaction between features, and therefore it can select redundant features.

The second group of filter methods aims at selecting a subset of features to be used for classification by considering the interaction between features within each evaluated candidate subset of features. For example, one of the most well-known multivariate filter feature selection methods is called Correlation-based Feature Selection (CFS) [48, 49, 137], which is based on the following hypothesis:

> *"A good feature subset is one that contains features highly correlated with (predictive of) the class, yet uncorrelated with (not predictive of) each other"*
> – Hall, 1999.

The approach used by the CFS method for evaluating the relevance (*Merit*) of a candidate subset of features based on the above hypothesis is based on Equation 2.9, which is based on Pearson's linear correlation coefficient (r) used for standardised numerical feature values. In Equation 2.9, k denotes the number

$$Merit_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{2.9}$$

of features in the current feature subset; $\overline{r_{cf}}$ denotes the average correlation between class and features in that feature subset; $\overline{r_{ff}}$ denotes the average correlation between all pairs of features in that subset. The numerator measures the predictive power of all features within that subset, which is to be maximised; while the denominator measures the degree of redundancy among those features in the subset, which is to be minimised.

Another part of CFS is the search strategy used to perform a search in the feature space. A lot of heuristic search methods have been applied, e.g. Hill-climbing search, Best First search and Beam search [103], and recently genetic algorithms [64, 65]. However, the CFS method based on genetic algorithms addresses the task of multi-label classification, where an instance can be assigned two or more class labels simultaneously, a more complex type of classification task which is out of the scope of this thesis.

The search strategy implemented in the Weka version of CFS, used in our experiments reported in other chapters is *Backward-Greedy-Stepwise*, which conducts a backward greedy search in the feature subset space. The termination criterion is when the deletion of any remaining feature leads to a decrease on validation results.

Another example of multivariate filter method is Markov Blanket-based feature selection [10, 42, 105, 132, 138, 139]. Given a Directed Acyclic Graph (DAG) where each node represents a variable, the Markov Blanket $\mathbf{M}_f$ for an individual feature f is defined as the set of all parent and child features of f, and the other features that are parents of f's child features. The features within $\mathbf{M}_f$ are the most relevant features with respect to f, since f is statistically independent from all other features outside the Markov Blanket given $\mathbf{M}_f$. As an example is shown in Figure 2.11, where only the nodes in black denote the features within the Markov Blanket of the *Class* attribute.

A well-known Markov Blanket discovery algorithm is *Incremental Association Markov Blanket (**IAMB**)* [114]. **IAMB** consists of two stages, namely the Grow stage and the Shrink stage. In the Grow stage, features which are outside the Markov Blanket will be considered to be added into the set of *Candidate Markov Blanket* (*CMB*), where some features will be removed at the Shrink stage. The construction of *CMB* starts from an empty set, then each feature will be heuristically evaluated whether its inclusion into the existing *CMB* maximises a heuristic

function $f(X;T|CMB)$, e.g. the mutual information, which measures the degree of relevance between feature X and the target attribute T given the set of features in the *CMB*. Before formally adding each candidate feature X into *CMB*, IAMB will check whether feature X and target feature T are not independent given *CMB*, mathematically shown as $\neg I(X;T|CMB)$. At the second stage (Shrink stage), **IAMB** removes in turn the features from *CMB* which are independent from T given *CMB* excluding those features, using the function $I(X;T|CMB-X)$.

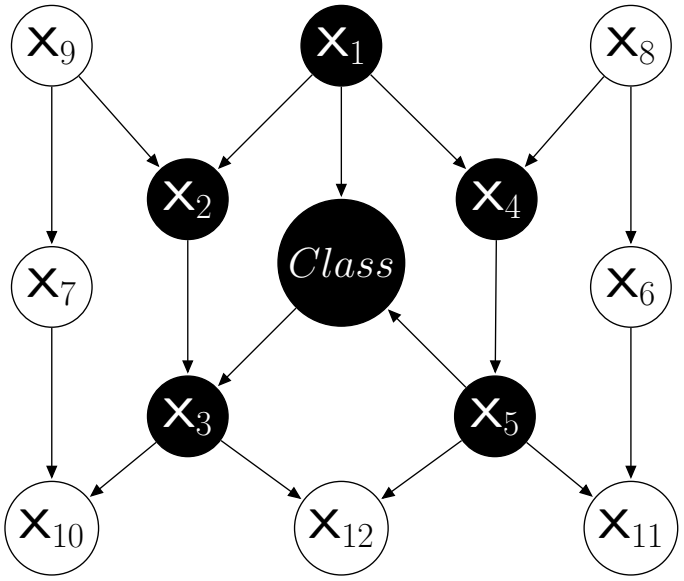


FIGURE 2.11 Example of the Markov Blanket for the *Class* Attribute

2.5.3 The Embedded Approach

Embedded feature selection methods conduct the feature selection process within the process of building the classifier, rather than conducting feature selection before building the classifier. As shown in Figure 2.12, between the stages of features input and accuracy report, the feature selection and the classifier building process are within the same stage.

For example, within the process of building a Decision Tree classifier, each feature is evaluated as a candidate for splitting the set of instances in the current tree node based on the values of that feature. Another example of embedded feature

selection method is linear regression methods, which will be discussed in the next section. In this thesis, we also proposed one embedded feature selection method for the Tree Augmented Naïve Bayes classifier, to be described in detail in Chapter 5.

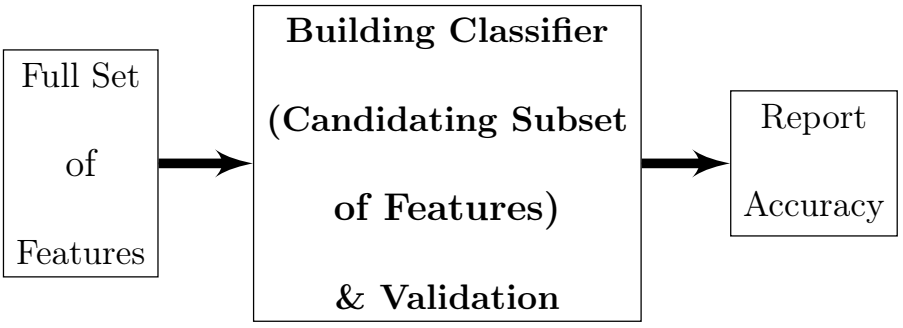


FIGURE 2.12 Flow-Chart of the Embedded Feature Selection Approach - Adapted from [84]

2.6 Hierarchical Feature Selection

Hierarchical feature selection methods are a specific type of feature selection methods based on the principle of exploiting the hierarchical relationships among features in order to improve the quality of the selected feature subset. This type of feature selection method is the *theme* of this thesis.

There has been very little research so far on hierarchical feature selection, i.e. on feature selection methods that exploit the generalisation-specialisation relationships in the feature hierarchy to decide which features should be selected. Hierarchical feature selection methods have been proposed for the task of selecting "enriched" Gene Ontology terms (terms that occur significantly more often than expected by chance) [9] and the task of learning linear models for regression, where the target variable to be predicted is continuous [59, 87, 134, 140]. Note that these tasks are quite different from the classification task addressed in this paper, where the goal is to predict the value of a categorical (or nominal) class variable for an instance based on the values of features describing properties of that instance. In any case, a brief review of these methods is presented next.

Alexa, et al. (2006) [9] proposed two methods to identify enriched Gene Ontology (GO) terms in a group of genes using the dependency information retained in the GO hierarchy. The first proposed method exploits the hierarchical dependencies between GO terms, i.e. the calculation of the p-value for each GO term starts from the bottom-most level of the GO Graph. If a GO term is found as significant based on its p-value, then all genes associated with that GO term's ancestor terms will be removed from that GO term's set of associated genes. This significance test will be applied until all GO terms have been processed.

The second method calculates the significance score of GO terms using the weights of their associated genes. The adjustment of weights for individual GO terms takes into account the significance score of its children GO terms. If the significance score for one child GO term is greater than the one for its parent GO term, then the weights for that parent term and all ancestor GO terms will be increased, and then the weight of that child GO term will also be re-computed. This adjustment process will be iteratively executed until there does not exist any child GO term whose weight is greater than any of its ancestor's weights. Both methods showed better performance than competing methods.

Another group of hierarchical feature selection methods is based on the Least Absolute Shrinkage and Selection Operator (LASSO) [52, 113], which is a linear regression method that performs embedded feature selection. In general, LASSO aims to find the parameters (regression coefficients) of a linear model that minimises both the value of a loss function and the value of a regularisation term, which penalises models with large values of feature weights. The need to minimise the value of the regularisation term forces the construction of sparse models, where many features with a weight of "0" are eliminated. Therefore, LASSO effectively selects a subset of relevant features.

Variations of the LASSO method perform hierarchical feature selection by using regularisation terms that consider the feature hierarchy. Briefly, a feature can be added into the set of selected features only if its parent feature is also included in that set. LASSO could be seen as one type of embedded feature selection method, since it removes features during the stage of model training. LASSO has been successful in various applications such as biomarker selection, biological network construction, and magnetic resonance imaging [134].

2.7 Hierarchical Redundancy

In this section, we described a type of redundancy which is a key concept for the feature selection methods proposed in later chapters. In this thesis, we define hierarchical redundancy as the situation where there exists more than one features that are related via a *specialisation-generalisation* relationship and have the same value (i.e. either "0" or "1"). In the example shown in Figure 2.13, the features can be grouped into two sets, i.e. a set of features having value "1" (the left four features: E, F, G, C), and another set of features having value "0" (the right four features: H, A ,B, D). In terms of features E, F, G, C, feature E is the parent of F, which is the parent of G. Feature G has the child C. It means that the value "1" of C logically implies the value "1" of G, whose value implies the value of F, and the value of F implies the value of E. Therefore, it can be noted that feature E is hierarchically redundant with respect to F, G and C; feature F is hierarchically redundant with respect to G and C; and feature G is hierarchically redundant with respect to C.

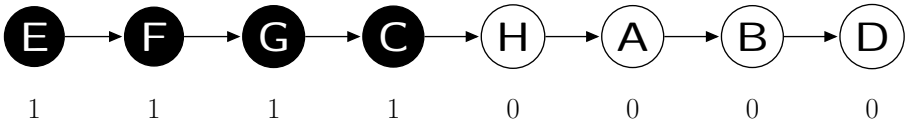


FIGURE 2.13 Example of a Set of Hierarchical Redundant Features

Analogously to the set of features having values "1", the other set of features having values "0" contains a similar type of hierarchical redundancy. In details, the value "0" of feature H logically implies the value "0" of A, whose value implies the value of B, and the value of B implies the value of D. Therefore, it can be noted that feature D is hierarchically redundant with respect to B, A and H; feature B is hierarchically redundant with respect to feature A and H; and feature A is hierarchically redundant with respect to H.

This type of hierarchical redundancy could be retained by a more complicated scenario, i.e. a given directed acyclic graph (DAG) structure of features. As shown in Figure 2.14, the DAG actually is composed by a set of different paths, where each individual path contains a set of hierarchically structured features. Note

that some features are shared by more than one path, e.g. feature F is shared by 4 paths, feature I is shared by 4 paths, feature A is shared by 3 paths, etc. This scenario of hierarchically structured features, with hierarchical redundancy as defined earlier, is the core problem addressed in this thesis, and we propose later feature selection methods that *remove or at least reduce the hierarchical redundancy among features.*

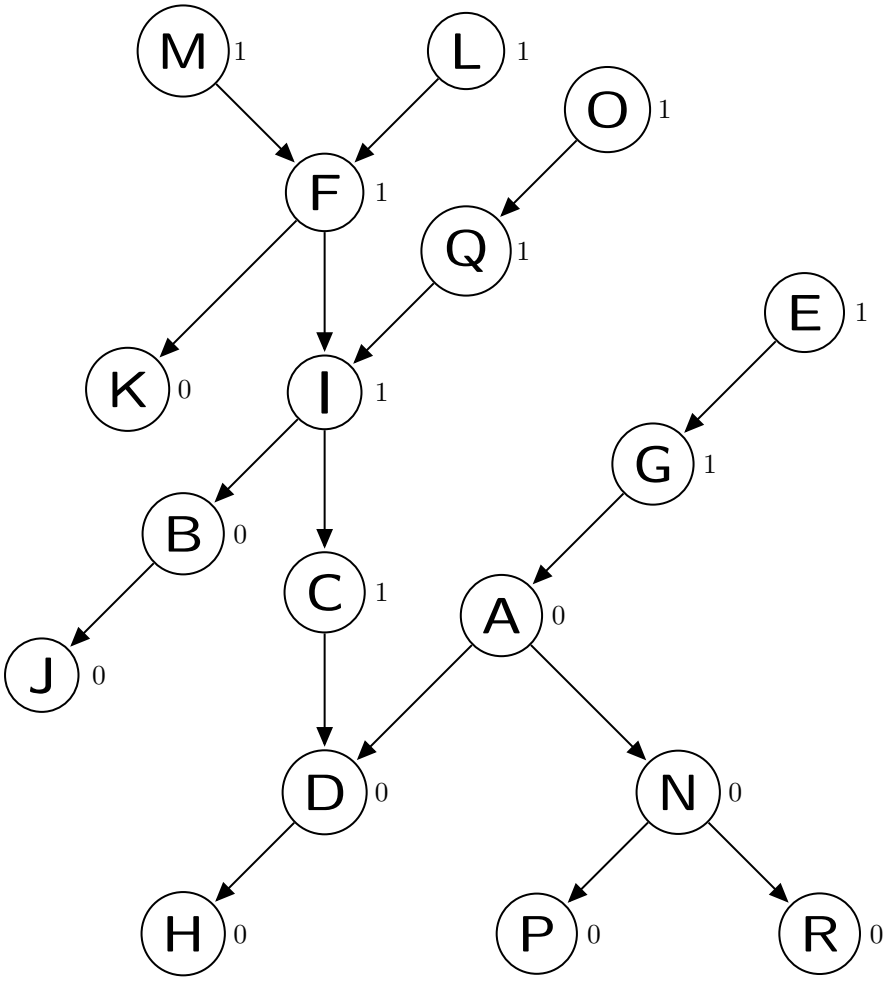


FIGURE 2.14 Example of a Set of Hierarchical Redundant Features Structured as a DAG

Note that this type of hierarchical redundancy scenario fits well with the lazy learning paradigm, i.e. the hierarchical redundancy occurs in the context of the values of features in an individual instance. For instance, Table 2.1 is an example dataset matrix, where each individual row represents one instance consisting of the value of the class attribute (in the last column) and the values of a set of features (in all other columns). The set of features in this example dataset matrix

retains the hierarchical dependencies associated with the feature DAG shown in Figure 2.14. For example, in the first row, the value of feature C equals to "1", then the values of features I, F, M, L are all equal to "1"; and *vice versa*, the value of feature A equals "0", then the values of features D and H are both equal to "0". Therefore, all proposed novel feature selection methods and the classifiers used in this thesis are based on the lazy learning scenario.

TABLE 2.1 Example Matrix of Dataset

	C	I	F	M	L	K	O	Q	B	J	...	A	D	H	*Class*
$Inst_1$	1	1	1	1	1	0	1	1	0	0	...	0	0	0	1
$Inst_2$	0	0	0	1	1	0	1	1	0	0	...	1	0	0	0
$Inst_3$	0	1	1	1	1	1	1	1	1	1	...	1	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
$Inst_n$	0	0	1	1	1	1	1	1	0	0	...	1	0	0	1

2.8 Final Remarks

The previous sections have reviewed background knowledge on data mining, specially about Bayesian network classification algorithms and feature selection methods. Recall that we decided to adopt Naïve Bayes, Tree Augmented Naïve Bayes (TAN) and Bayesian Network Augmented Naïve Bayes (BAN) as the classification algorithms for the research in this thesis. The reasons are described next.

First, considering the interpretability of classifiers, "white-box" classifiers are more suitable for the theme of this thesis, i.e. conducting data mining and knowledge discovery from ageing-related data, in order to discover knowledge or patterns that can be interpreted by biologists. The models learnt by Naïve Bayes, TAN and BAN are in principle more interpretable than the "black-box" models built by classification algorithms like Support Vector Machines and Artificial Neural Networks [38].

Second, considering that the theme of this thesis focuses on hierarchical feature selection methods, there are good reasons to expect that such methods can improve the predictive performance of Bayesian network classifiers, as follows. Bayesian network classifiers are sensitive to redundant features [18] and the proposed hierarchical feature selection methods are designed to eliminate or at least reduce redundancy among features, as discussed in later chapters. In addition, some Bayesian network classifiers like BAN does not scale well with the large number of features, due to the overfitting problem, i.e. there is a large number of parameters that need to be learnt from the training dataset, but those learnt parameters might not work well on the testing dataset.

Another reason for focusing on Bayesian network classifiers is due to the learning approach used by this type of classifiers, in terms of the distinction between eager and lazy learning paradigms discussed in Section 2.2.4. In this thesis, the proposed hierarchical feature selection algorithms follow the lazy learning paradigm, i.e. the proposed algorithms conduct feature selection for each individual testing instance and then lazy learning-based classifiers are used for classifying the individual instance based only on the selected features. Naïve Bayes, TAN and BAN can be naturally adapted for working under the lazy learning paradigm, as will be shown in Chapters 4, 5 and 6.

Hierarchical feature selection is the main research theme for this thesis. As described in Chapters 4, 5 and 6, we propose and evaluate four hierarchical feature selection methods and two methods for constructing the network topology of a BAN classifier. These methods have been shown, overall, to improve the predictive performance of Bayesian network classifiers.

Chapter 3

Background on the Biology of Ageing and Bioinformatics

3.1 Introduction

Ageing is an ancient research topic that has attracted scientists' attention for a long time, not only for its practical implications on extending the longevity of human beings, but also due to its high complexity.

With the help of modern biological science, it is possible to start to reveal the mysteries of ageing. In this thesis, we focus on research about the biology of ageing, which is an application topic associated with our proposed hierarchical feature selection methods, which will be described in the next three chapters.

In this chapter, we will briefly review basic concept of molecular biology; biology of ageing; and bioinformatics.

3.2 Overview of Molecular Biology

Molecular Biology is defined by the *Oxford Dictionary* as "the branch of biology that deals with the structure and function of the macromolecules essential to life". More precisely, molecular biology focuses on understanding the interactions between DNA, RNA and proteins, including the regulation of the systems consisting

of those macromolecules.

Such regulation mechanisms include the process of *gene expression*, which can

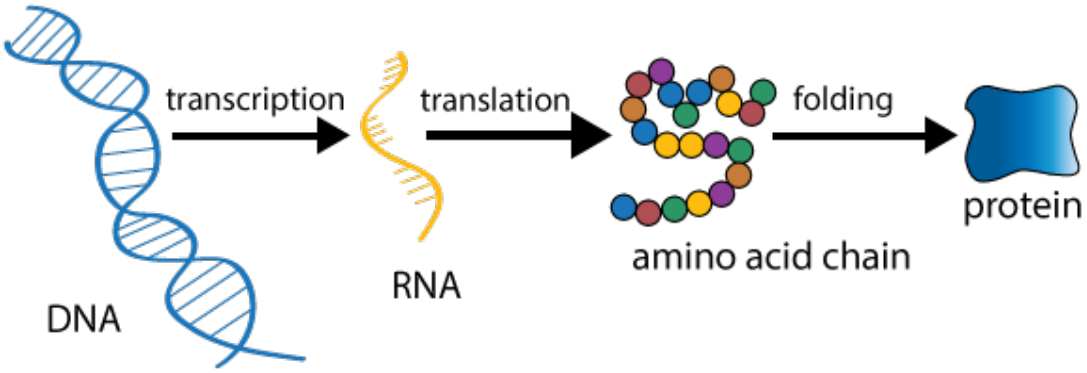


FIGURE 3.1 Overview of the Gene Expression Process [117]

be divided into three main stages, i.e. transcription, translation and protein folding as shown in Figure 3.1. At the stage of transcription, Deoxyribonucleic acid (DNA), which is a type of nucleic acid that contains the genetic information, is transcribed into messenger RNA (mRNA), then the mRNA will be translated into the amino acid sequence of a protein, which is finally folded into a 3D structure in the cell.

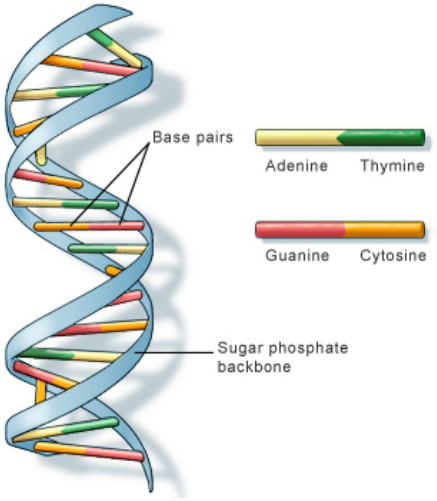


FIGURE 3.2 DNA Double Helix [119]

The basic units of DNA consist of adenine (A), guanine (G), cytosine (C) and thymine (T), and a DNA sequence can be represented by the combination of A, G, C, and T, such as ATAAGCTC [115]. The 3D structure of DNA is a double helix

(Figure 3.2), where one strand governs the synthesis of a complementary RNA molecule during the transcription process [102].

RNA, which is another type of nucleic acid, plays an important role on the process of protein production. RNA has basic units that are the same units of DNA with the exception that thymine (T) in DNA is replaced by uracil (U) in RNA. The structure of RNA is represented as a chain of nucleotides, which is different from DNA having a double helix structure. There exist different types of RNA, e.g. mRNA, tRNA, rRNA, etc. Among those types of RNA, mRNA performs its function during the stage of transcription, which is defined as the synthesis of RNA based on a DNA template [115] or the process of copying one of the DNA strands into an RNA [102]. Then the next step is translation, by which the linear sequence of information retained in mRNA is decoded and used for producing linear chains of amino acids, which are the basic component for proteins and determine the structure of proteins [102].

A gene is considered as a segment/unit of DNA containing heredity information and defines particular characteristics/functions of proteins [102]. As shown in Figure 3.3, gene-1 and gene-2 are respectively contained by different segments of DNA, which is stored in a chromosome. Briefly, one specific gene controls different functions of proteins, and therefore affects particular functions of organisms, such as the effect on the metabolism rate, which is possibly an ageing-related factor that will be discussed later.

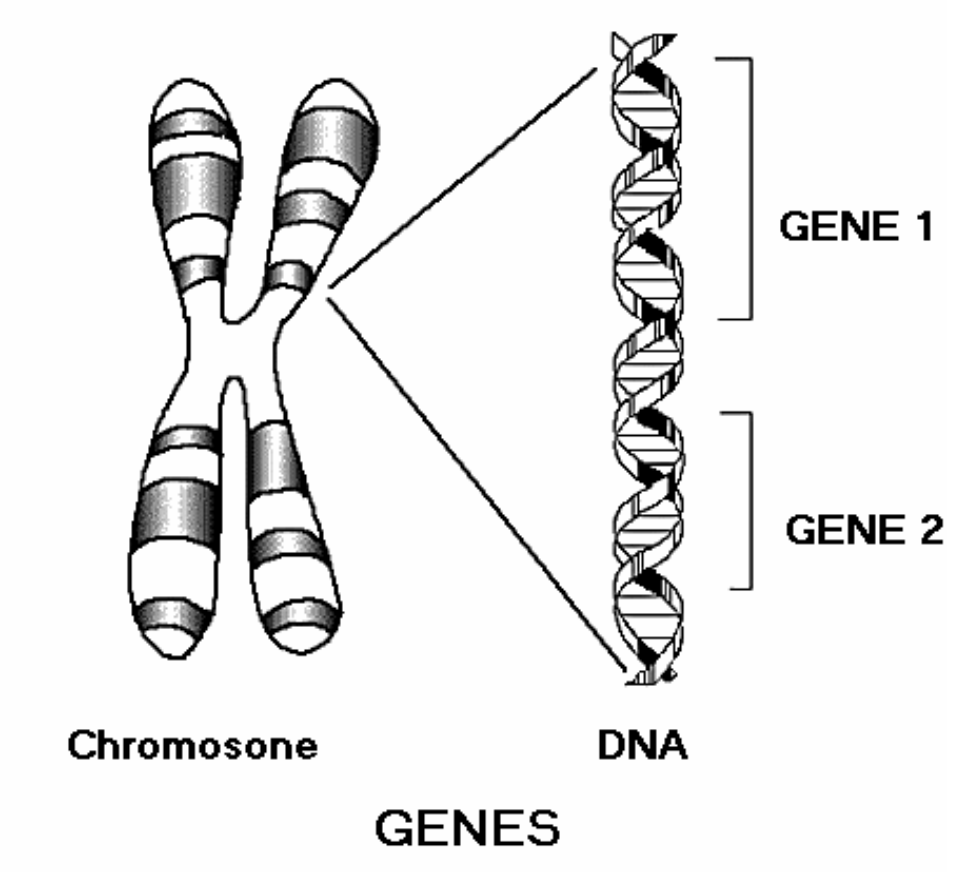
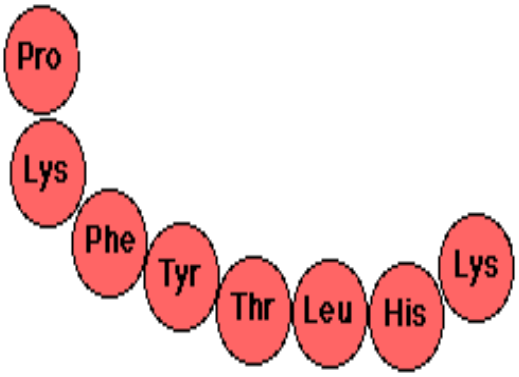


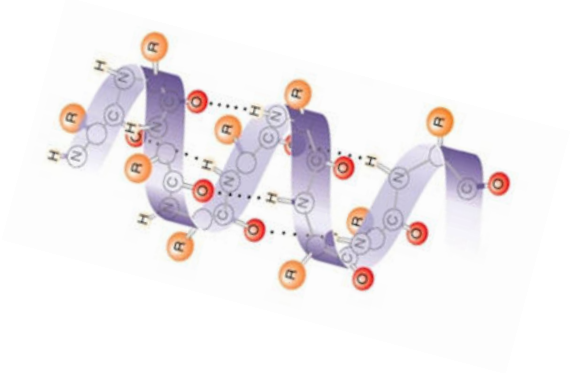
FIGURE 3.3 Example of Genes within DNA [118]

Proteins are large biological molecules that carry out almost all of living cells' functions, most of which are determined by the ability of proteins to recognize other molecules through binding [19]. The functions of proteins can be categorised into three major broad groups: structural proteins, which are considered as the organism's basic building blocks; enzymes, which regulate biochemical reactions; and transmembrane proteins that maintain the cellular environment [21].

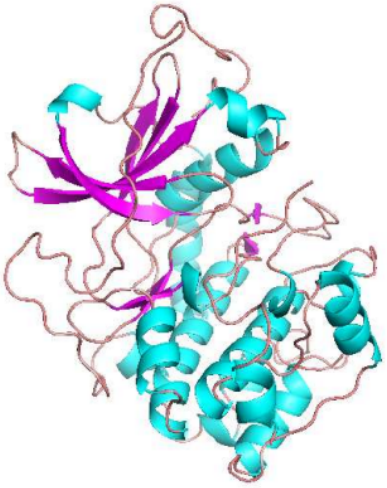
Proteins consist of 20 different types of amino acids that are joined together to compose a linear sequence named poly-peptide chain [21]. Proteins have four types of structure (Figure 3.4). The primary structure (Figure 3.4.A) is a linear amino acid sequence which determines all other three types of structures. The secondary structure consists of α helices (Figure 3.4.B) and β sheets. The tertiary structure (Figure 3.4.C) is a 3D structure that is built according to the spontaneous folding of poly-peptides in the cell environment. It is made by α helices, β sheets, other minor secondary structures and connecting loops [115]. The quaternary structure (Figure 3.4.D) is composed by two or more poly-peptide chains with the same



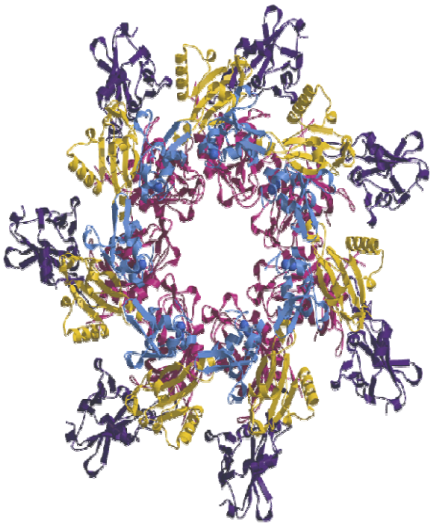
(A) Primary Structure



(B) Secondary Structure



(C) Tertiary Structure



(D) Quaternary Structure

FIGURE 3.4 Protein Structures [1–3]

forces that stabilise tertiary structure [115].

In this thesis, we focus on ageing-related genes. Recall that one specific gene controls certain functions for organisms by producing certain proteins. We will review some factors associated with ageing, including some discovered age-related genes and their related biological processes in the next section.

3.3 Overview of the Biology of Ageing

3.3.1 Introduction to the Biology of Ageing

Ageing is a complex and stochastic process of progressive function loss for an organism with time [71], and the accumulation of function losses leads to the

mortality of the organism. The speed of ageing and the longevity of organisms differs between species. For example, *C. elegans'* lifespan is around 2-3 weeks [66], whereas the ocean quahog has 400 years of longevity. In terms of human longevity, the longest age record is 122.5 years and the average longevity measured in 2009 was 79.4 years in the UK [128].

The mystery of ageing is a sophisticated issue that has puzzled humans for thousands of years, as there has been many stories about a failure on finding the method of being immortal. Nowadays, with the help of molecular biology, some possible factors related to ageing have been found, as discussed next.

3.3.2 Some Possible Ageing-Related Factors

Some ageing-related factors have been revealed with the help of molecular biology, such as genetic factors, environmental factors, etc. From the perspective of molecular biology, those factors have an effect on ageing through their regulation of ageing-related biological pathways.

A biological pathway is a series of actions among molecules in a cell that leads to a certain product or a change in a cell [57]. Biological pathways analysis is considered as an approach to research the molecular mechanisms of ageing. In particular, the pathways related with the regulation of growth, energy metabolism, nutrition sensing and reproduction seem associated with the process of ageing [120].

Genetic factors have been shown to be one of the most important types of factor that impacts on biological pathways related with the ageing process. The mutation of a gene(s) change(s) the effects of pathways on organisms. For instance, it has been found that a gene called *daf-2* is highly related to the extension of lifespan in *C.elegans* (a worm). The mutation of *daf-2* will affect the activation of FOXO proteins that can activate cell maintenance and stress resistance mechanisms [68]. In addition, changes on *daf-2* are related with insulin/insulin-like growth factor-1 (IGF-1) signaling. The former is a hormone that regulates the metabolism of glucose and the latter primarily controls growth [120]. It was found that inhibiting insulin/IGF-1 signaling or increasing the activity of FOXO extends *Drosophila's* lifespan [68]. Conversely, it was found that mutations that increase oxidative damage can shorten lifespan. For example, the *ctl-1* mutants shorten lifespan and prevent lifespan extension of *daf-2* mutants by age-associated lipofuscin granules

accumulation [46]. Therefore, it is possible to speculate that gene mutations, especially changes on the sensitivity of the insulin/IGF-1 receptor, can enhance the resistance to environmental stress [68]. In support of this inference, the relationship between stress responsiveness and lifespan was found for *age-1* mutants in *C. elegans* [32]. In addition, this point of view is also supported by another possible ageing-related pathway, i.e. the target of rapamycin (TOR) pathway. TOR kinase stimulates growth and blocks salvage pathways [68] that are related with autophagy (a basic repair mechanism for damaged cell degradation), which can alleviate the accumulation of damages on cells.

Oxidative stress was found as an ageing-related factor. In essence, the role of oxidative stress on longevity regulation is related with reactive oxygen species (ROS), which are a type of byproduct of normal metabolism [101]. It was discovered that the balance between ROS and an antioxidant defence system controls the degree of oxidative stress, which is associated with modifications of cellular proteins, lipids and DNA [32]. Also, other research revealed that a cycle of growing DNA damage is caused by damaged mitochondria, which leads to increased ROS production [32]. ROS can damage and crosslink DNA, proteins and lipids [120] and affect the formation of base adducts of mutation and canceration-related DNA [53]. Therefore, the damage caused by oxidation reactions, cell or DNA self-repair mechanisms and resistance to environmental stress are probably interacting factors that affect the process of ageing, and all of them are supported by the theory that the reduction of energy intake associated with calorie restriction (discussed next) will be helpful for extending longevity.

Nutritional level is another type of environment factor. This was discovered in 1935 by McCay, Crowell and Maynard [90] under well-executed studies, which discovered that the longevity of rats can be extended by a dietary control approach. Then several findings showed that the method of dietary control for extending longevity can be applied to other species, such as yeast, fish, hamster, etc. [88]. Caloric restriction was found to be helpful for extending lifespan with the possible reason of oxidative damage attenuation. The joint impact of reduced rate of reactive oxygen molecules generation and increased efficiency of protective processes might alleviate the accumulation of oxidative damages; the evidence for this was found in isolated mitochondria and microsomes from caloric restricted rodents [88].

In addition, some diseases (in particular, most types of cancers) are also factors

that are highly related with ageing. Cancer cells could be seem as immortal, and this is opposite to normal cells that have intrinsic process of senescence. Some research revealed that cell senescence might be a mechanism of tumour suppression [116]. The experiments about observing the function of *p53* (a gene that prevents cancer) supported that hypothesis. Finkel, Serrano and Blasco (2007) [33] found that mice which over-expressed *p53* could be resistant to cancer, but was found as prematurely aged; and reduction of *p53* expression prevents telomere- or damage-induced senescence [22]. The possible reasons would be due to the fact that *p53* helps to avoid or reduce genomic instability, which is considered the hallmark of both cancer and ageing. However, the relationship between ageing and cancer is very complex and has not been precisely understood so far.

3.3.3 The Evolutionary History Theory of Ageing

The evolutionary history theory of ageing is a popular explanation about the difference of longevity between species. Firstly, the natural selection principle plays an essential role on the development of a species' lifespan. The rate of ageing will be concomitantly changed with changes on the force of natural selection [71]. Especially in hazardous environments, the surviving individuals would promote their somatic maintenance ability and propagate their gene variants [120]. Also, a deleterious mutation will not be easily passed to offspring via reproduction, since the effect of a mutation usually appears in early life [43], before the individual has a chance to reproduce. On the other hand, if a mutation has a deleterious effect that occurs only in late life, long after the organism has reproduced, there is little selection pressure to eliminate that kind of mutation (since it does not affect the reproduction of the organism).

Secondly, the competition between species will suppress the growth of longevity expectation for the weaker, as limited resources would not support the energy consumption in harsh environmental conditions [70]. The weaker competitor usually could not have enough time for evolution. For example, the observation on a mainland population and an island population of *Didelphis virginiana* revealed that the latter has longer longevity, since they have reduced exposure to predators comparing with the former [13]. The evolutionary history hypothesis provides

a macro-perspective about the development of lifespan expectation for different species.

3.3.4 Mysteries in Ageing Research

Although some findings about the possible reasons for the process of ageing have been revealed, several mysteries about ageing still cannot be figured out. To start with, the actual biological mechanisms leading to ageing are still not clear. For example, the actual function of longevity-associated genes with respect to the stress resistance is unknown [32] and the answer about how different ageing-related biological pathways interact and cooperate is still absent [120]. Moreover, it is not clear how gene mutations affect ageing-related cellular degeneration [120]. Furthermore, the diversity between species limits the universality of support from those hypotheses about the reasons of ageing. In terms of the caloric restriction theory, which caloric restriction approach extends the lifespan and the actual molecular mechanism underlying that extension are still debated, and whether caloric restriction extends longevity in long-lived species is unknown [53]. Therefore, discovering answers to the mysteries of ageing is challenging, as the vast variety of ageing-related factors interactively work, and the answers are still a long way to go.

3.4 An Overview of Protein/Gene Function Prediction in Bioinformatics

3.4.1 Introduction to Bioinformatics

Bioinformatics is an inter-disciplinary field that integrates computer science, mathematics, statistics, etc., with the purpose of assisting biological research. Bioinformatics can be defined as follows:

> *"The science of collecting and analysing complex biological data such as genetic codes."* - Oxford Dictionary

The main subareas of bioinformatics consist of biological data management, biological data analysis software development and research on biological data analysis methods.

In terms of biological data management, there exists a lot of biological databases with different types of biological data. For example, the well-known *GenBank* database is a collection of publicly available nucleotide sequences [16]; the Biological General Repository for Interaction Datasets (*BioGRID*) is a repository of data about physical and genetic interactions from model organisms [109]; and *REACTOME* is a curated database about human pathways and reactions [25]. Those bioinformatics databases foster the development of bioinformatics and also promote biology research, since the biological data in these databases are well stored, integrated or managed.

Based on those biological databases, a lot of applications have been made for supporting biology research, e.g. protein/gene function prediction [20, 39, 77, 100, 104], protein structure prediction [14, 55, 62, 63, 74], etc. In this thesis, the main theme is developing novel biological data analysis methods, in particular novel feature selection methods for the classification task of data mining, and using them for predicting a kind of gene function; more precisely for predicting the effect of a gene on the longevity of an organism.

3.4.2 Protein/Gene Function Prediction

As one of the main tasks in bioinformatics, protein function prediction has been highly valued due to its advantages of saving time and reducing cost, since it can be used for guiding the direction of biological experiments designed to confirm whether a protein has a certain function. A biologist can conduct only experiments focusing on fewer specific proteins whose function have been predicted with high confidence, rather than conducting a large amount of slow and expensive biological experiments. The methods for gene/protein function prediction can be categorised into three main broad groups, i.e. sequence alignment analysis, 3D structure similarity analysis, and machine learning/data mining methods. We will review those three groups of methods in the next three subsections.

3.4.2.1 Sequence Alignment Analysis Methods

Sequence Alignment Analysis is the most conventional approach to predict the functions of proteins and genes. A well-known Sequence Alignment Analysis-based method, named Basic Local Alignment Search Tool (BLAST), has been highly valued and widely applied on protein/gene function prediction. The basic principle of BLAST is measuring the degree of similarity between the amino acid sequence of a protein with unknown function and the amino acid sequence of a set of proteins with known functions. The motivation for this approach is that a protein's amino acid sequence dictates the protein's 3D structure, which further determines the function of the protein. In this approach, an unknown-function protein is predicted to have the functions of its most similar known-function proteins.

In details, BLAST employs a measure of local similarity called maximal segment pair (MSP) score between two sequences and also detects whether the score will be improved by extending or shortening the segment pair by using a dynamic programming algorithm [12]. Then a user-defined threshold is used for filtering the most reliable MSPs. Based on this basic principle, BLAST has been extended for fitting more applications, such as Primer-BLAST [133], IgBLAST [135], etc.

Although BLAST has dominated in the area of protein/gene function prediction, it has several limitations, as follows [36]. Firstly, BLAST is only applicable for predicting the function of proteins or genes which are similar to known-function proteins/genes. Secondly, similar amino acid sequences do not guarantee similar functions between proteins, because of the difference of their 3D structure. Therefore, the high score obtained by BLAST might not be quite reliable. Thirdly, in the context of coping with hierarchical protein function data, such as the data consisting of generalisation-specialisation relationships used in this thesis (discussed later), BLAST has the limitation of ignoring such hierarchical relationships.

3.4.2.2 3D Structure Analysis-Based Protein Function Prediction

In a cell, the folds of proteins will spontaneously change depending on cellular environment factors. Therefore, it is uncertain that a high degree of similarity between amino acid sequences will lead to similar functions. In general, the information

about protein structure is more valuable in terms of protein function prediction. The second group of methods for protein function prediction is based on protein 3D structure analysis. There are some protein folds that are associated with multiple functions, but most folds have been found to represent a unique function [39]. Some algorithms based on the knowledge of folds don't fit the expectation of high accuracy. For the purpose of overcoming that shortage, a more reliable strategy consisting of analysing the structure patterns of proteins that are spatial regions within protein structure, denoting unique markers for specific functions, has been proposed [39].

The basic concept of a 3D structure analysis-based protein function prediction algorithm consists of two parts: 3D motif library generation and a searching algorithm for matching motifs between two proteins [39]. For example, a well-known 3D structure analysis-based protein function prediction server *ProFunc* [77] detects the possible function of unknown proteins by using a graph-matching algorithm to compare the secondary structure elements (SSEs) between target proteins and the proteins whose SSEs are known and stored in the databases. In addition, *ProFunc* further analyses the cleft size, residue type and other details of structural information about the protein.

3D structure analysis has attracted attention due to its highly reliable predictive results. There are several tools based on structure analysis that are available to be used by the bioinformatics community, such as SuMo, PINTS, PDBFun, etc.

3.4.2.3 The Machine Learning/Data Mining Approach

Machine learning/data mining methods have been widely applied in bioinformatics research, such as in the task of protein/gene function prediction. Unlike the popular sequence similarity-based methods, such as BLAST, the machine learning/data mining approach can be called a model induction or alignment-free approach. Briefly, this approach treats protein function prediction as a classification task, where the protein functions are classes and the predictor attributes (or features) are properties or characteristics of protein. One of the advantages of machine learning/data mining-based protein function prediction methods (more precisely, classification methods) is that they can predict the functions of a given protein without being given existing similar proteins (i.e. protein with amino

acid sequence similar to the protein being classified). More precisely, classification methods take into account the variables (attributes) denoting different types of biological properties that might be associated with protein function prediction.

A lot of different types of classifiers have been adopted for different tasks of protein/gene function prediction and have shown powerful predictive performance. For example, Support Vector Machine (SVM), which is a type of classifier that obtains very good predictive performance in general, have been widely used. For instance, Borgwardt, et al. (2005) [20] classified proteins into functional classes by applying SVM with graph kernels; Bhardwaj, et al. (2005) [17] used SVM to predict DNA-binding proteins; and Krishnan and Westhead (2003) [75] applied SVM and Decision Tree classifiers to predict the effects of single nucleotide polymorphisms on protein function. Note, however, that SVMs have the disadvantage of producing "black-box" classification models, which in general cannot be interpreted by biologists.

Bayesian network classifiers are another group of classifiers that are widely applied in protein function prediction, due to their advantage of producing probabilistic graphic models that can be interpreted by biologists. For example, Yousef, et al. (2007) [136] used Naïve Bayes to predict microRNA targets. As another example, Barutcuoglu, et al. (2006) [15] proposed to use a Bayesian network to cope with the prediction inconsistency problem that happens in a hierarchical classifier. Inconsistent hierarchical predictions occur, e.g. when a classifier predicts for a given instance, a certain class y, but not an ancestor of class y in the hierarchy. This is inconsistent, assuming the class hierarchy is a "is-a" hierarchy, so that an instance assigned to a class must be assigned to its ancestor classes. That Bayesian network calculates the most probable prediction results by Bayes' theorem. More specifically, they trained an individual SVM classifier for each class, so that the different SVMs can make inconsistent predictions across the class hierarchy, and then combined the predictions of all those SVMs by using a Bayesian network.

Apart from classifiers, feature selection methods also play an important role on protein function prediction, due to their capacity of improving the predictive performance of classifiers by providing the classification algorithm with a subset of very relevant features, removing features with little relevance or containing redundant information for classification purposes. For example, Glaab, et al. (2012) [45] adopted three different types of eager learning-based feature selection algorithms,

i.e. partial least squares-based feature selection (PLSS), correlation-based feature selection and random forest-based feature selection, working with rule-based evolutionary machine learning systems to tackle the microarray data classification task. The experimental results show that PLSS outperforms other non-univariate feature selection methods and indicate that the feature independence assumption could be beneficial for microarray gene selection tasks. Note that those three types of feature selection methods select a feature subset for classifying all testing instances, following the eager learning paradigm. Al-Shahib, et al. (2005) [7] adopted a type of wrapper feature selection method with a genetic search algorithm combined with SVM, Decision Tree and Naïve Bayes classifiers for predicting protein functions for the *Neisseria gonorrhoea* proteome. In another work of Al-Shahib, et al. (2005) [8], they proposed a new feature selection approach. This feature selection approach first ranks all features according to those features' corresponding p-values calculated by the Wilcoxon rank sum test between each feature and the class variable, and then removes the redundant features with respect to the features from top to the bottom of the ranking table. The method used to detect redundancy is based on the correlation coefficient. Li, et al. (2012) [81] adopt the mRMR (minimal-redundancy-maximal-relevance) method [94] to select the optimal subset of features for predicting protein domain. This method firstly ranks all features according to the quality measure computed by the mRMR method, and then evaluates the predictive performance of different subsets of features by stepwise adding one feature into the current feature subset. The adding order is from high to low on the features' ranking. In addition, Leijoto, et al. (2014) [80] adopted genetic algorithms to select a subset of physical-chemical features to predict protein functions.

3.4.3 A Comparison Between Three Approaches for Protein/Gene Function Prediction

Comparing machine learning/data mining methods and sequence alignment analysis methods, the latter seems to have more limited reliability in general. As mentioned in the previous section, although the primary structure broadly determines the functions of proteins, it is also possible that two proteins have different

functions while their primary structure are quite similar. That means the high score obtained by sequence alignment will not guarantee a high degree of similarity between the functions of the aligned proteins/genes. For example, according to research on Gene Ontology term annotation errors, the error rate of annotation inferred by sequence similarity reaches 49% in same cases [36]. In addition, the sequence alignment methods have the drawback of not discovering relationships between biochemical properties and protein functions, which would be valuable for biologists.

Comparing machine learning/data mining methods and 3D structure analysis methods, the latter show high accuracy in terms of protein function prediction. However, the obvious limitation of 3D structure analysis methods is that there are many proteins whose 3D structure is unknown. Therefore, in the case of predicting functions of an unknown protein, the prediction method's accuracy is limited by the availability of proteins that not only have a known 3D structure, but also have a 3D structure similar to the current unknown protein.

Although machine learning/data mining methods show advantages of flexibility and potential for discovering comprehensible models, compared with the other two methods, the model induction approach also has the limitation of not producing comprehensible models sometimes, when the choice of data mining algorithm(s) is not appropriate. More precisely, as an advantage of black-box classifiers, their high predictive accuracy attracts most researchers' attention in the bioinformatics community. Especially, artificial neural networks and support vector machines are widely used as protein function prediction methods. However, as mentioned earlier, in general, those classifiers cannot be interpreted by users and they cannot reveal valuable insight on relationships between protein features (properties) and protein function. Therefore, white-box (interpretable) classifiers, such as Bayesian network classifiers, Decision Trees, etc., should receive more attention in area of protein function prediction.

3.5 Related Work on The Machine Learning/Data Mining Approach Applied to Biology of Ageing Research

There exist few works about the machine learning/data mining approach with application on ageing-related proteins/genes function prediction. As the key application area of this PhD project, the use of classification methods for predicting the functions of ageing-related proteins/genes has been investigated by the bioinformatics community only in the last few years, so there is a broad space for research in this area. The relevant articles in this research topic are briefly reviewed as follows.

Freitas, et al. (2011) [35] addressed the classification of DNA repair genes into ageing-related or non-ageing related by applying conventional data mining techniques on datasets which consisted of ageing-related protein/gene data and several types of features. The experiments revealed that protein-protein interaction information, which was obtained from the HPRD (Human Protein Reference Database) [98], is helpful for prediction. Other predictor features, such as biological process Gene Ontology (GO) terms, evolutionary gene change rate, and types of DNA repair pathway were used for the prediction task. After comparing the results of two different classification algorithms, Naïve Bayes outperformed J48 (a Decision Tree algorithm) in terms of predictive accuracy. But with the help of the J48 algorithm, some interesting and interpretable IF-THEN rules which can be used for classifying a DNA repair gene into an ageing-related gene or a non-ageing-related gene were found. Similarly, Fang, et al. (2013) [30] addressed the classification of ageing-related genes into DNA repair or non-DNA repair genes. Both studies used GO terms as features, in addition to other types of features.

GO terms are particularly relevant for this thesis, since they are the type of feature to which the feature selection methods proposed in this thesis were applied. Hence, GO terms will be discussed separately in the next section.

Li, et al. (2010) [82] classified *C. elegans* genes into longevity and non-longevity

genes by adopting a support vector machine (SVM). They firstly created a functional network by adopting information about gene sequences, genetic interactions, phenotypes, physical interactions and predicted interactions from wormnet [79]. Then they derived graph features from the functional network, such as a node's degree, longevity neighbour ratio, average shortest distance, etc. The experiments showed that the predictor features as a whole contribute to a high predictive accuracy, up to 85%.

Huang, et al. (2012) [54] proposed a method using the information about the effect of a gene's deletion on lifespan to predict whether the deletion of a specific gene will affect the organism's longevity. The three effect classes were: no effect on lifespan, increased or decreased lifespan. They adopted network features, biochemical and physicochemical features, and functional features obtained from the deletion network, which was constructed by mapping the information about gene deletion and protein-protein interaction data (obtained from the STRING database [60]). For each deleted gene, they removed its downstream lifespan-related genes from the complete lifespan-related gene network and considered the remaining network as the deletion network for that gene. They computed GO enrichment scores (based on the p-value of a hypergeometric test) as functional features of the deletion networks. A two-layer classifier was used to firstly detect whether the deletion of one gene will affect the longevity, then another classifier predicts the specific function of that gene in terms of longevity.

These works regarding ageing-related gene classification/prediction shed a light on ageing-related knowledge discovery based on data mining approaches. However, given the small number of works in this research topic, there is still much space for further research, not only in terms of optimising the predictive accuracy, but also finding new clues that help to solve or reduce the mystery of ageing, by discovering knowledge that can be interpreted by biologists.

3.6 Biological Databases Relevant to This Research

In this section, we discuss the two biological databases used in our research, i.e. the Gene Ontology and the Human Ageing Genomic Resources (HAGR).

3.6.1 The Gene Ontology (GO)

The Gene Ontology project aims to provide dynamic, structured, unified/controlled vocabularies for the annotation of genes [112]. To minimise the inconsistent annotations of individual genes between different biological databases, it is required that a centralised public resource provides universal access to the ontologies, annotation datasets and software tools. In addition, an ontology can facilitate communication during research cooperation and improve the interoperability between different systems. The initial members/contributors of the Gene Ontology Consortium were FlyBase, *Saccharomyces* Genome Database and the Mouse Genome Informatics project, whereas now the number of databases members rose to around 36. The information resources of GO consist of documentation-supported links between database objects and GO terms with the experimental evidence from the published literature for individual source information, in order to provide high-quality GO annotations. In addition, the standard for GO term annotation defined that all GO terms should not be species specific.

There are three categories of GO terms, each implemented as a separate ontology: biological process, molecular function, and cellular component [112]. The biological process represents a biological objective to which a gene product contributes, such as regulation of DNA recombination, regulation of mitotic recombination, etc. The process might be accomplished by one or more assemblies of functions. Note that the meaning of a biological process is not necessarily consistent to the meaning of a biological pathway. The molecular function ontology represents the biochemical level of gene functions, regardless of the location or when that function occurs, such as lactase activity. The cellular component refers to a location where the gene product is active, such as ribosome, nuclear membrane, etc.

In terms of structure of the GO information, there are hierarchical relationships between GO terms. The hierarchical relationships are composed mainly by "is-a" relationships, which is the type of hierarchical relationship considered in this research. That is, the process, function or location represented by a GO term is a specific instance of the process, function or location represented by its parent GO term(s). Hence, these hierarchical relationships are effectively generalisation-specialisation relationships. Examples of such hierarchical relationships are shown

in the example graph in Figure 3.5 where GO:0051234 (establishment of localization) and GO:0044699 (single-organism process) are both a child of GO:0008150 (biological process), and GO:0006810 (transport) is a child of GO:0051234 and a parent of GO:0044765, which is a child of not only GO:0006810, but also GO:0044699, and also a parent of GO:0045056 (transcytosis). These hierarchical relationships can be used for building a *Directed Acyclic Graph* (DAG) composed by GO terms.

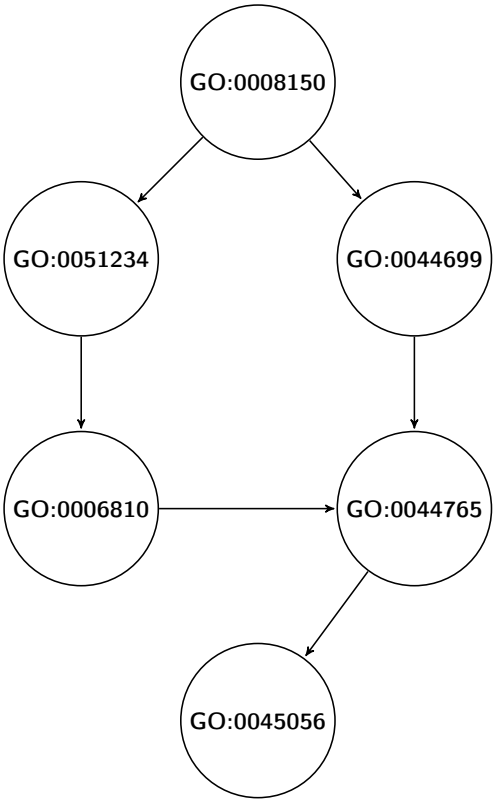


FIGURE 3.5 Example of a Topology of Gene Ontology Data

3.6.2 Human Ageing Genomic Resources (HAGR)

The HAGR is a high-quality biological database that specifically focuses on the biology or genetics of ageing. The HAGR database consists of four main groups of data, namely GenAge, AnAge, GenDR and DAA (Digital Ageing Atlas).

Firstly, GenAge is a database of ageing/longevity-associated genes for humans and model organisms, such as mice, worms, fruit flies and yeast. GenAge includes high-quality curated information of genes that have been shown to have noticeable effect on changes in the ageing phenotype and/or longevity [27]. GenAge

consists of three sections, i.e. 1) a set of ageing-associated genes for human, 2) a set of longevity-associated genes for model organisms, 3) a set of mammalian genes whose expression is commonly altered during ageing in multiple issues.

Secondly, AnAge is a database that focuses on animal ageing and longevity. The reason for building this database is providing sufficient data that can be used for conducting comparative analysis on ageing mechanisms between different species. AnAge contains longevity-related data about 4,205 species, which consists of mammals, birds, reptiles, amphibians and fishes in version of *Build 12* [111]. The data included in AnAge is of high quality and confidence, based on data from authoritative sources and checked by curators.

Thirdly, HAGR includes GenDR, which is a database designed for the analysis of how caloric restriction extends lifespan, consisting of data about dietary restriction-essential genes, which are defined as those genes that interfere with dietary restriction lifespan extension after being genetically modified, but do not have impact on the lifespan of animals under the condition of an *ad libitum* diet [27]. In addition, as complementary information, GenDR includes a set of mammalian genes differentially expressed under dietary restriction condition.

In addition, DAA is a centralised collection of human ageing-related changes that integrates data from various biological levels, e.g. molecular, cellular, physiological, etc [24]. DAA provides a system-level and comprehensive platform for ageing research, focuses on ageing-associated changes.

Overall, GenAge offers a bioinformatics platform where ageing-associated genes can be found through a user-friendly interface, and is a way of integrating information about ageing-related genes, for the purpose of functional genomics and systems biology analysis. Also, as an overall picture of ageing-associated genes, GenAge provides sufficient data for conducting data mining research, which will be discussed in a later section and is the application theme of this project.

Chapter 4

Lazy Hierarchical Feature Selection Methods with Naïve Bayes

4.1 Introduction

In this chapter we describe three proposed hierarchical feature selection methods, namely *Select Hierarchical Information-Preserving (HIP) Features*, *Select Most Relevant (MR) Features* and the hybrid *Select Hierarchical Information-Preserving and Most Relevant (HIP–MR) Features*. In this chapter these methods are used to select features, in a data pre-processing phase, for the Naïve Bayes classification algorithm. These methods will also be used to select features for the Tree Augmented Naïve Bayes (TAN) algorithm in Chapter 5. All these hierarchical feature selection methods work in the scenario of lazy learning (discussed in Chapter 2, i.e. feature selection is performed separately for each testing instance). The hierarchical feature selection methods described in this chapter, as well as part of the computational results reported here, have been published in [122, 123].

4.2 Select Hierarchical Information-Preserving (HIP) Features

The *Select Hierarchical Information-Preserving (HIP) Features* method focuses only on eliminating the hierarchical redundancy in the set of selected features, ignoring the relevance values of individual features. Recall that two features are hierarchically redundant, in a given instance, if they have the same value in that instance and are located in the same path from a root to a leaf node in the feature graph (for more details on hierarchical redundancy, see Chapter 2). The motivation for eliminating the hierarchical redundancy among selected features is that some types of classification algorithms, like Naïve Bayes, are particularly sensitive to redundancy among features, as discussed earlier.

The pseudocode of the HIP method is shown as Algorithm 4.1, where **TrainSet** and **TestSet** denote the training dataset and testing dataset, and they consist of all input features; $\mathbb{A}(x_i)$ and $\mathbb{D}(x_i)$ denote the set of ancestors and descendants (respectively) of the feature x_i; $Status(x_i)$ means the selection status ("Selected" or "Removed") of the feature x_i; $\mathbf{Inst}_{<w>}$ means the current instance being classified in **TestSet**; $Value(x_{i,w})$ denotes the value of feature x_i ("1" or "0") in that instance; A_{ij} denotes the jth ancestor of the feature x_i; D_{ij} denotes the jth descendant of the feature x_i; **TrainSet_FS** denotes the shorter version of the training dataset where all features' status are "Selected"; and $\mathbf{Inst\_FS}_{<w>}$ denotes the shorter version of instance w that consists only of features whose status is "Selected".

In the first part of Algorithm 4.1 (lines: 1 – 8), it firstly constructs the **DAG** of features, finds all ancestors and descendants of each feature in the DAG, and initialises the status of each feature as "Selected". During the execution of the algorithm, some features will have their status set to "Removed", whilst other features will remain with their status set "Selected" throughout the algorithm's execution. When the algorithm terminates, the set of features with status "Selected" is returned as the set of selected features.

In the second part of Algorithm 4.1 (lines: 9 – 27), it performs feature selection for each testing instance in turn, using a lazy learning approach. For each instance, for each feature x_i, the algorithm checks its value in that instance. If x_i has value

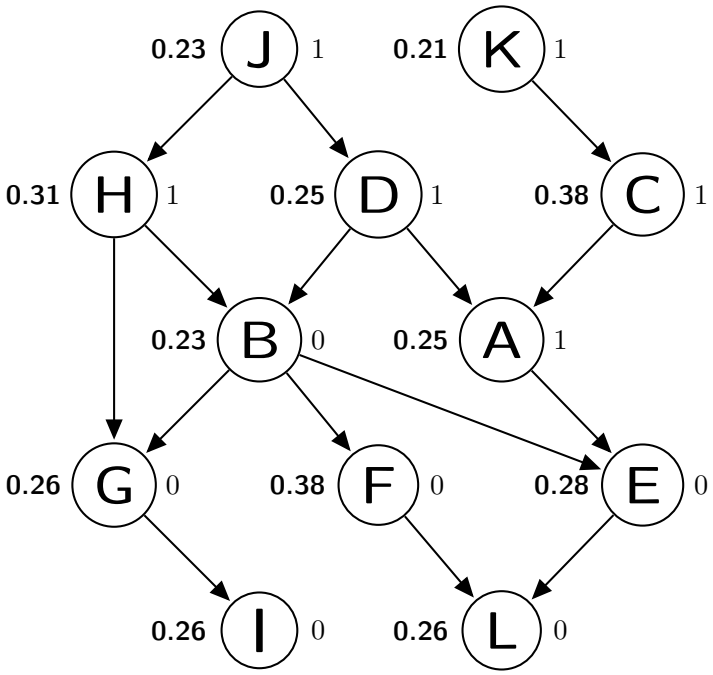


FIGURE 4.1 Example of a Small DAG of Features

"1", all its ancestors in the DAG have their status set to "Removed" – since the value "1" of each ancestor is redundant, being logically implied by the value "1" of x_i. If x_i has value "0", all its descendants have their status set to "Removed" – since the value "0" of each descendant is redundant, being logically implied by the value "0" of x_i.

To show how the second part of Algorithm 4.1 works, we use as example a hypothetical testing instance with just 12 features, denoted by the letters A – L. Figure 4.1 shows a small hypothetical DAG specifying the hierarchical relationships among the features of our hypothetical instance. In Figure 4.1, the relevance and value ("1" or "0") for each feature is shown on the left (in bold) and on the right (respectively) of the node representing that feature. Note that the HIP feature selection method uses only information about the feature values and their hierarchical relationships; the features' relevance values are used only by the two other feature selection methods described later.

With respect to the example DAG in Figure 4.1, lines 10 – 20 of Algorithm 4.1 work as follows. When feature A is processed, the selection status of its ancestor features D, J, C and K will be assigned as "Removed" (lines: 12 – 14), since the value "1" of A logically implies the value "1" of all of A's ancestors. Analogously, when feature B is processed, the selection status of its descendant features G, I,

F, L and E will be assigned as "Removed" (lines: 16 – 18), since the value "0" of B logically implies the value "0" of all of B's descendants. When feature C (with value "1") is processed, its ancestor K has its status set to "Removed". And so on, processing one feature at a time.

Note that the status of a feature may be set to "Removed" more than once, as it happened for feature K in the earlier example. However, once the status of a feature is set to "Removed", it cannot be re-set to "Selected" again. Hence, the result of Algorithm 4.1 does not depend on the order in which the features are processed.

After processing all features in the example DAG, the features selected by the loop in lines 10 - 20 are A, B and H. Note that these three *core* features contain the complete hierarchical information associated with all the features in the DAG of Figure 4.1, in the sense that the observed values of these three core features logically imply the values of all other features in that DAG.

Next, the training dataset and current testing instance are reduced to contain only features whose status are "Selected" (lines: 21 – 22), and that reduced instance is classified by Naïve Bayes (line: 23). Finally, the status of all features is reassigned as "Selected" (lines: 24 – 26), as a preparation for feature selection for the next testing instance.

Algorithm 4.1 Select Hierarchical Information-Preserving (HIP) Features

```
Initialize DAG with all features in Dataset;
Initialize TrainSet;
Initialize TestSet;
for each feature x_i do
    Initialize 𝔸(x_i) in DAG;
    Initialize 𝔻(x_i) in DAG;
    Initialize Status(x_i) ← "Selected";
end for
for each Inst_<w> ∈ TestSet do
    for each feature x_i ∈ DAG do
        if Value(x_{i,w}) = 1 then
            for each ancestor A_ij ∈ 𝔸(x_i) do
                Status(A_ij) ← "Removed";
            end for
        else
            for each descendant D_ij ∈ 𝔻(x_i) do
                Status(D_ij) ← "Removed";
            end for
        end if
    end for
    Re-create TrainSet_FS with all features x_i where Status(x_i) = "Selected";
    Re-create Inst_FS_<w> with all features x_i where Status(x_i) = "Selected";
    NaïveBayes(TrainSet_FS, Inst_FS_<w>);
    for each feature x_i do
        Re-assign Status(x_i) ← "Selected";
    end for
end for
```

4.3 Select Most Relevant (MR) Features

The *Select Most Relevant (MR) Features* method performs feature selection considering both the relevance values of individual features and the hierarchical redundancy among features. Like the HIP method, for each feature x_i in the current instance being classified, MR first identifies the set of features whose values are implied by the value of x_i in that instance – i.e. either the ancestors of x_i, if x_i has value "1"; or the descendants of x_i, if x_i has value "0", for each path from the current node to a root or a leaf node of the feature DAG, depending on whether the current feature has value "1" or "0", respectively. Next, MR compares the relevance of x_i and all features in each identified path. Among all those features (including x_i), MR marks for removal all features, *except* the most relevant feature. If there are more than one features with the same maximum relevance value in a given path, as a tie-breaking criterion, MR retains the most specific (deepest) feature among the set of features with value "1" or the most generic (shallowest) feature among the set of features with value "0" – since those features' values logically imply the largest number of other features' values, among the set of features being compared.

As a part of our feature selection method, we use Equation 4.1 to measure the relevance ($\mathbf{R}$), or predictive power of a binary feature x_i taking value x_{i1} or x_{i2},

$$\mathbf{R}(x_i) = \sum_{c=1}^{n} [\mathbf{P}(y_c|x_{i1}) - \mathbf{P}(y_c|x_{i2})]^2 \tag{4.1}$$

where y_c is the c-th class and n is the number of classes. A general form of Equation 4.1 was originally used in [108] in the context of Nearest Neighbour algorithms, and here it has been adjusted to be used as a feature relevance measure for feature selection algorithms. In this work, n=2, x_i is a feature, and Equation 4.1 is expanded to Equation 4.2, where the two terms being added in the right part of the equation are equal, as shown in Theorem 4.1, followed by the corresponding proof.

$$\begin{aligned} \mathbf{R}(x_i) = {} & [\mathbf{P}(y{=}1 \mid x_i{=}1) - \mathbf{P}(y{=}1 \mid x_i{=}0)]^2 \\ & + [\mathbf{P}(y{=}0 \mid x_i{=}1) - \mathbf{P}(y{=}0 \mid x_i{=}0)]^2 \end{aligned} \tag{4.2}$$

Equation 4.2 calculates the relevance of each feature as a function of the difference in the conditional probabilities of each class given different values ("1" or "0") of a feature, indicating whether or not a instance is annotated with that feature.

Theorem 4.1. *In Equation 4.1,*

if $n = 2$*, so that* $\mathbf{R}(x_i) = [P(y_1|x_{i1}) - P(y_1|x_{i2})]^2 + [P(y_2|x_{i1}) - P(y_2|x_{i2})]^2$*,*

we have: $[P(y_1|x_{i1}) - P(y_1|x_{i2})]^2 = [P(y_2|x_{i1}) - P(y_2|x_{i2})]^2$.

***Proof*:**

$$
\begin{aligned}
\because\ & [P(y_1|x_{i1}) + P(y_2|x_{i1}) = 1] \wedge [P(y_1|x_{i2}) + P(y_2|x_{i2}) = 1] \\
\therefore & [P(y_1|x_{i1}) - P(y_1|x_{i2})]^2 = [(1 - P(y_2|x_{i1})) - (1 - P(y_2|x_{i2}))]^2 \\
& = [1 - P(y_2|x_{i1}) - 1 + P(y_2|x_{i2})]^2 \\
& = [-P(y_2|x_{i1}) + P(y_2|x_{i2})]^2 \\
& = [-(P(y_2|x_{i1}) - P(y_2|x_{i2}))]^2 \\
& = [P(y_2|x_{i1}) - P(y_2|x_{i2})]^2 \qquad \square
\end{aligned}
$$

The pseudocode of the MR method is shown as Algorithm 4.2, where $\mathbf{R}(x_i)$ denotes the value of relevance for the ith feature; $\mathbb{A}_+(x_{i,k})$ and $\mathbb{D}_+(x_{i,k})$ denote the set of features containing both the ith feature and its ancestors or descendants (respectively) in the k-th path; **MRF** denotes the most relevant feature among the set of features in $\mathbb{A}_+(x_{i,k})$ or $\mathbb{D}_+(x_{i,k})$; $A_{i,j,k+}$ and $D_{i,j,k+}$ denotes the j^{th} feature in $\mathbb{A}_+(x_{i,k})$ and $\mathbb{D}_+(x_{i,k})$, respectively.

In the first part of Algorithm 4.2 (i.e. lines 1 – 9), firstly the **DAG** will be constructed, then $\mathbb{A}_+(x_{i,k})$ and $\mathbb{D}_+(x_{i,k})$ for each feature x_i at each path k will be initialized, and the relevance ($\mathbf{R}$) value for each feature will be calculated. In the second part of the algorithm (i.e. lines 10 – 34), the feature selection process will be conducted for each testing instance using a lazy learning approach.

To show how the second part of Algorithm 4.2 works, we use again as example the DAG shown in Figure 4.1. When feature A (with value "1") is processed (lines: 13 – 18), the features in two paths, i.e. path (a) containing features J, D and A; and path (b) containing features K, C and A, are processed. In path (a), the features having maximum relevance value are D and A; but only feature A is selected as the **MRF** (line: 14), since it is deeper than feature D in that path. In path (b), only feature C is selected as **MRF**, since it has the maximum relevance value. Hence, after processing feature A, all features contained in the two paths have their status set to "Removed", except feature C (lines: 15 – 17).

Analogously, when feature B (with value "0") is processed, the features in three paths, i.e. path (a) containing features B, G and I; path (b) containing features B, F and L; and path (c) containing features B, E and L will be processed. In path (a), both features G and I have maximum relevance value, but G will be selected as the **MRF** (line: 21) since it is shallower than I. In path (b), feature F is selected as the **MRF** since it has the maximum relevance value among all features in that path. In path (c), feature E is selected as the **MRF**, since it also has the maximum relevance value. Therefore, after processing feature B, the selection status for all features contained in those three paths will be assigned as "Removed", except features G, F and E (lines: 22 - 24).

After processing all features in that example DAG, the selected features are H, C, G, F and E. Next, the training dataset and the current testing instance are reduced to contain only those five selected features in line 28 - 29 of Algorithm 4.2, and that reduced instance is classified by Naïve Bayes in line 30. Finally, the status of all features is reassigned to "Selected" in line 31 – 33, as a preparation for feature selection for the next instance.

Note that, for each set of features being compared when MR decides which features will have their status set to "Removed", this decision is based both on the relevance values of the features being compared and the hierarchical redundancy among features, as explained earlier. Thus, in general the MR method does not select all core features with complete hierarchical information on feature values, as selected by HIP (see Section 4.2). Consider, e.g. the core feature B = "0", which implicitly contains the hierarchical information that features G, I, F, L and E have value "0". Also, the core feature A = "1" implies that features D, J, C and K have value "1". The features B and A were selected by the HIP method, but neither B

nor A is selected by the MR method, because the relevance value of B is smaller than the relevance values of G, F and E; and the relevance value of A is smaller than the relevance value of feature C. Hence, we lose the information about the values of nodes B and A, whose values are not implied by the values of features G, F, E and C (nor implied by any other feature in the DAG).

On the other hand, the MR method has the advantage that in general it selects features with higher relevance values than the features selected by the HIP method (which ignores feature relevance values). For instance, in the case of our example DAG in Figure 4.1, the three features selected by HIP (A, B and H) have on average a relevance value of 0.263, whilst the five features selected by MR (H, C, G, F and E) have on average a relevance value of 0.322.

Algorithm 4.2 Select Most Relevant (MR) Features

Initialize **DAG** with all features in Dataset;
Initialize **TrainSet**;
Initialize **TestSet**;
for each feature x_i on path k in **DAG do**
    Initialize $\mathbb{A}_+(x_{i,k})$ in **DAG**;
    Initialize $\mathbb{D}_+(x_{i,k})$ in **DAG**;
    Initialize $Status(x_i) \leftarrow$ "Selected";
    Calculate $\mathbf{R}(x_i)$ in **TrainSet**;
end for
for each $\mathbf{Inst}_{<w>} \in$ **TestSet do**
    for each feature $x_i \in$ **DAG do**
        if $Value(x_{i,w}) = 1$ **then**
            for each path k from x_i to root in **DAG do**
                Find **MRF** in $\mathbb{A}_+(x_{i,k})$;
                for each ancestor $A_{i,j,k+}$ except **MRF do**
                    $Status(A_{i,j,k+}) \leftarrow$ "Removed";
                end for
            end for
        else
            for each path k from x_i to leaf in **DAG do**
                Find **MRF** in $\mathbb{D}_+(x_{i,k})$;
                for each descendant $D_{i,j,k+}$ except **MRF do**
                    $Status(D_{i,j,k+}) \leftarrow$ "Removed";
                end for
            end for
        end if
    end for
    Re-create **TrainSet_FS** with all features x_i where $Status(x_i) =$ "Selected";
    Re-create $\mathbf{Inst\_FS}_{<w>}$ with all features x_i where $Status(x_i) =$ "Selected";
    NaïveBayes(**TrainSet_FS**, $\mathbf{Inst\_FS}_{<w>}$);
    for each feature x_i **do**
        Re-assign $Status(x_i) \leftarrow$ "Selected";
    end for
end for

4.4 Select Hierarchical Information-Preserving and Most Relevant (HIP–MR) Features

Although both HIP and MR select a set of features without hierarchical redundancy, HIP has the limitation of ignoring the relevance of features, and MR has the limitation that it does not necessarily select all core features with the complete hierarchical information (features whose observed values logically imply the values of all other features for the current instance). The hybrid *Select Hierarchical Information-Preserving and Most Relevant (HIP–MR) Features* method addresses these limitations, by both considering feature relevance (like MR) and selecting all core features with the complete hierarchical information (like HIP). The price paid for considering both these criteria is that, unlike HIP and MR, HIP–MR typically selects a large subset of features having some hierarchical redundancy (although less redundancy than the original full set of features), as will be discussed later.

For each feature x_i in the instance being classified, HIP–MR first identifies the features whose values are implied by the value of x_i in the instance – i.e. the set of features which are ancestors or descendants of x_i, depending on whether x_i has value "1" or "0", respectively. Then, HIP–MR removes features by combining ideas from the HIP and MR methods, as follows. If feature x_i has value "1", HIP–MR removes the ancestors of x_i whose relevance values are not greater than the relevance value of x_i. If feature x_i has value "0", HIP–MR removes the descendants of x_i whose relevance values are not greater than the relevance value of x_i.

Therefore, HIP–MR selects a set of features where each feature has the property(ies) of being needed to preserve the complete hierarchical information associated with the instance being classified (the kind of feature selected by HIP) or has a relatively high relevance in the context of its ancestors or descendants (the kind of feature selected by MR). Hence, the set of features selected by the HIP–MR method tends to include the union of the sets of features selected by the HIP and MR methods separately, making HIP–MR a considerably more "inclusive" feature selection method.

The pseudocode is shown as Algorithm 4.3. In the first part of the algorithm (lines: 1 – 9), firstly the **DAG** is constructed, the ancestors and descendants of

each feature are found, and the relevance value of each feature is calculated by Equation 4.1. In the second part of the algorithm (lines: 10 – 32), the feature selection process is carried out by combining ideas of the HIP and MR methods, as explained earlier, for each testing instance, following a lazy learning approach.

In the case of our example feature **DAG** in Figure 4.1, when feature A (with value "1") is processed, its relevance value is compared with the relevance values of all its ancestor features J, D, C and K. Then, features J, D and K are marked for removal, since their relevance values are not greater than the relevance of A. Next, when feature B (with value "0") is processed, none of its descendant features is marked for removal, since their relevance values are greater than the relevance value of B. This process is repeated for all other features in the instance being classified. At the end of this process, the selected features are: H, C, B, A, G, F and E.

Note that in this example HIP–MR selects all features selected by HIP or MR. Actually, as will be shown in Section 4.6.1, HIP–MR tends to select substantially more features than the number of features selected by HIP and MR together. Note also that, although HIP–MR selects a feature subset with less hierarchical redundancy than the original full feature set, the features selected by HIP–MR still have some redundancy, unlike the features selected by HIP and MR. This is because HIP–MR can select a redundant feature x_i if x_i has higher relevance than another selected feature logically implying x_i. For instance, in the above example, HIP–MR selects feature C, which is redundant with respect to selected feature A, since C has higher relevance than A.

Algorithm 4.3 Select Hierarchical Information-Preserving and Most Relevant (HIP–MR) Features

```
Initialize DAG with all features in Dataset;
Initialize TrainSet;
Initialize TestSet;
for each feature x_i in DAG do
    Initialize 𝔸(x_i) in DAG;
    Initialize 𝔻(x_i) in DAG;
    Initialize Status(x_i) ← "Selected";
    Calculate R(x_i) in TrainSet;
end for
for each Inst_<w> ∈ TestSet do
    for each feature x_i ∈ DAG do
        if Value(x_{i,w}) = 1 then
            for each ancestor A_ij ∈ 𝔸(x_i) do
                if R(A_ij) ≤ R(x_i) then
                    Status(A_ij) ← "Removed";
                end if
            end for
        else
            for each descendant D_ij ∈ 𝔻(x_i) do
                if R(D_ij) ≤ R(x_i) then
                    Status(D_ij) ← "Removed";
                end if
            end for
        end if
    end for
    Re-create TrainSet_FS with all features x_i where Status(x_i) = "Selected";
    Re-create Inst_FS_<w> with all features x_i where Status(x_i) = "Selected";
    NaïveBayes(TrainSet_FS, Inst_FS_<w>);
    for each feature x_i do
        Re-assign Status(x_i) ← "Selected";
    end for
end for
```

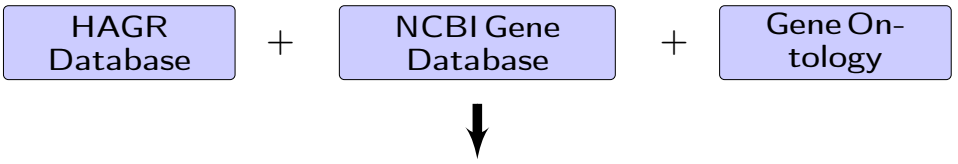
4.5 Experimental Methodology

4.5.1 Dataset Creation

We constructed four datasets with data about the effect of genes on an organism's longevity, by integrating data from the Human Ageing Genomic Resources (HAGR) GenAge database (Build 16) [27] and the Gene Ontology (GO) database (version: 2013-08-07) [112]. HAGR provides longevity-related gene data for four model organisms, i.e. *Caenorhabditis elegans*, *Saccharomyces cerevisiae*, *Drosophila melanogaster* and *Mus musculus*. We created one dataset for each of these model organisms. To begin with, the data from the HAGR database contains, as one of the identifiers for each gene, the *EntrezID*, which is adopted as the unique key for mapping from the HAGR data to the *gene2go* file [4], which contains information about GO terms associated with each gene. Then the integrated dataset created by retrieving data from the HAGR database and the *gene2go* file has been merged with the data from the GO database for the purpose of obtaining the relationship between each GO term and its ancestor GO terms. In addition, an iterative method had been implemented in order to collect all ancestor GO terms for each gene in the dataset; i.e. for each GO term associated with a gene, we get that GO term's parent GO term(s), then the parent(s) of that parent GO term(s), etc., until the root GO term (*note that the root GO term will not be included in the created dataset, due to its uselessness for prediction*). The structure of the newly created dataset is represented as shown in Figure 4.2, where the feature value "1" means the occurrence of a GO term with respect to each gene. In the class variable, the values of "Pro" and "Anti" mean "pro-longevity" and "anti-longevity". Pro-longevity genes are those whose decreased expression (due to knock-out, mutations or RNA interference) reduces lifespan and/or whose overexpression extends lifespan; accordingly, anti-longevity genes are those whose decreased expression extends lifespan and/or whose over-expression decreases it [111].

The GO terms that have only one associated gene would be useless for building a classification model because they are extremely specifically related to an individual gene, and the model that includes these GO terms would be confronted with the over-fitting problem. However, in terms of biological information contained in

GO terms, those GO terms associated with only a few genes might be valuable for discovering knowledge, since they might represent specific biological information. Therefore, it is necessary to investigate what is the most appropriate GO term frequency threshold for filtering the dataset through computational experiments. We will consider the thresholds in range of 3 – 10, taking into account the reliability of the classification model. It is necessary to check whether we would miss valuable knowledge involving very specific GO terms after we decide to adopt a higher threshold in order to avoid the over-fitting problem. The Gene Ontology consists of three main types of terms, i.e. Biological Process, Molecular Function and Cellular Component (with three corresponding root terms). Since the experiments with all different GO term frequency thresholds are very time consuming, these experiments will merely use the Biological Process GO terms. Further experiments using all three types of GO terms will be discussed in Section 4.7.2. In addition, our datasets do not include the GO term that is extremely general, occurring in all genes; i.e. GO: 0008150 (biological process), which is the root for all the GO terms of this type.



Gene\GO	GO_1	GO_2	GO_3	GO_4	...	GO_n	Class
Gene_1	1	0	0	1	...	0	Pro
Gene_2	0	1	0	0	...	1	Anti
Gene_3	0	0	0	1	...	1	Pro
...	...	...	...	...	...	...	...
Gene_n	1	0	1	0	...	0	Pro

FIGURE 4.2 Structure of the Created Dataset

TABLE 4.1 Detailed Information about the Created Datasets

	Caenorhabditis elegans	***Saccharomyces cerevisiae***	***Drosophila melanogaster***	***Mus musculus***
Initial Number of GO Terms	1528	1708	1595	2625
Initial Number of Instances	566	293	121	89
Number (%) of Pro-Longevity Instances	203 (35.9 %)	41 (14.0 %)	81 (66.9 %)	63 (70.8 %)
Number (%) of Anti-Longevity Instances	363 (64.1 %)	252 (86.0 %)	40 (33.1 %)	26 (29.2 %)

Additional information about the initial created datasets is shown in Table 4.1. The initial number of GO terms is the number of GO terms (features) in the dataset before removing GO terms with frequency of occurrence below a user-defined threshold and before running the feature selection methods.

4.5.2 Predictive Accuracy Measure

Generally, in our datasets, the distribution of instances belonging to the two classes is imbalanced, as shown in Table 4.1. Hence, we evaluate the predictive performance of classifiers by using the value of the Geometric Mean (GMean) between Sensitivity (Sen.) and Specificity (Spe.), defined as $\mathbf{GMean} = \sqrt{Sen. \times Spe.}$, because it takes into account the balance of the classifiers' Sen. and Spe. [58]. Sensitivity means the proportion of pro-longevity (positive class) genes that were correctly predicted as pro-longevity, and specificity means the proportion of anti-longevity (negative class) genes that were correctly predicted as anti-longevity in the testing dataset [11]. For all classifiers evaluated in this work, the reported

values of Sen., Spe. and GMean were computed by a well-known 10-fold cross validation procedure [129].

4.6 Results for Naïve Bayes Varying GO Term Frequency Thresholds

4.6.1 Experimental Results

We firstly report results comparing the GMean of four versions of Naïve Bayes (NB), namely standard-NB (without using any feature selection method) and HIP+NB, MR+NB and HIP–MR+NB, which denote NB applied on the set of features selected by the respective hierarchical feature selection method (HIP, MR or HIP–MR). The results are shown in Tables 4.2 – 4.5, where the bold figures denote the highest GMean value in the corresponding dataset version for each value of the GO term frequency threshold. The figures after "±" are standard errors.

In details, for the results about *Caenorhabditis elegans* in Table 4.2, the values of specificity are greater than the values of sensitivity obtained by all algorithms. MR+NB obtains the highest GMean value 6 out of 8 times, while HIP+NB obtains the highest GMean value the other two times. In Table 4.3, for the results about *Drosophila melanogaster*, the values of sensitivity are greater than the values of specificity obtained by all algorithms. HIP+NB obtains 6 out of 8 times the highest GMean value, and MR+NB obtains the highest value two times. Analogous to Table 4.3, the values of sensitivity are greater than the values of specificity obtained by all algorithms shown in Table 4.4. HIP+NB obtains 6 out of 8 times the highest GMean value, while MR+NB obtains the highest value two times (with one draw of highest GMean value to HIP+NB), and NB without feature selection obtains one time the highest GMean value. For the results about *Saccharomyces cerevisiae* in Table 4.5, the values of specificity are greater than the values of sensitivity. MR+NB obtains 5 out of 8 times the highest GMean value, while NB without feature selection obtains two times the highest GMean value, and HIP+NB obtains the highest value one time.

In terms of average GMean value among all dataset versions for the four model organisms, MR+NB obtained the highest value, i.e. 61.9%, which is slightly higher than HIP+NB's value, i.e. 61.6%. In terms of predictive performance on individual model organisms, MR+NB obtained the highest GMean value (averaged over all threshold values) in the *Caenorhabditis elegans* and *Saccharomyces cerevisiae* datasets; and it obtained the second highest GMean value in the other two datasets. Conversely, HIP+NB obtained the highest average GMean value in the *Drosophila melanogaster* and *Mus musculus* datasets; and it obtained the second highest GMean value in the *Caenorhabditis elegans* dataset. In summary, both MR+NB and HIP+NB have been successful feature selection methods, obtaining better results than both the baseline standard Naïve Bayes (without feature selection) and the HIP–MR+NB feature selection method.

TABLE 4.2 Sensitivity (%), Specificity (%) and Geometric Mean (%) of Hierarchical Feature Selection Methods with Naïve Bayes Classifier for *Caenorhabditis elegans* Datasets

Caenorhabditis elegans Datasets													
	Standard-NB			**HIP+NB**			**MR+NB**			**HIP–MR+NB**		
Thre.	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***
T3	52.6 ± 3.0	63.2 ± 4.4	57.7	53.7 ± 3.7	70.8 ± 3.5	61.7	57.9 ± 3.1	68.8 ± 4.7	**63.1**	51.6 ± 3.4	69.4 ± 3.8	59.8
T4	55.8 ± 3.9	61.8 ± 2.5	58.7	54.2 ± 3.9	71.2 ± 2.3	62.1	57.4 ± 4.0	68.8 ± 2.0	**62.8**	51.1 ± 5.0	68.1 ± 2.2	59.0
T5	54.7 ± 2.9	61.5 ± 3.9	58.0	52.1 ± 3.0	72.6 ± 2.4	61.5	57.4 ± 2.0	72.2 ± 3.0	**64.4**	51.6 ± 2.7	69.4 ± 3.8	59.8
T6	60.0 ± 4.6	59.4 ± 4.0	59.7	53.7 ± 4.6	71.9 ± 2.9	62.1	57.9 ± 2.9	69.8 ± 3.5	**63.6**	56.3 ± 4.8	67.4 ± 3.9	61.6
T7	56.8 ± 4.7	61.5 ± 3.1	59.1	55.8 ± 4.7	71.2 ± 2.7	**63.0**	55.3 ± 3.6	71.2 ± 2.6	62.7	56.3 ± 3.5	66.0 ± 2.3	61.0
T8	56.8 ± 4.8	56.8 ± 2.2	56.8	52.1 ± 5.2	70.0 ± 2.5	60.4	53.2 ± 3.4	70.4 ± 3.4	**61.2**	56.8 ± 4.3	62.4 ± 2.9	59.5
T9	57.9 ± 4.0	59.2 ± 4.0	58.5	51.1 ± 5.0	69.0 ± 3.0	**59.4**	48.9 ± 3.3	71.4 ± 3.3	59.1	53.2 ± 3.6	62.4 ± 4.2	57.6
T10	58.4 ± 4.8	57.1 ± 2.4	57.7	48.4 ± 3.6	69.7 ± 2.8	58.1	50.0 ± 3.5	71.4 ± 1.4	**59.7**	52.1 ± 4.1	61.3 ± 2.2	56.5
Ave.	56.6 ± 4.1	60.1 ± 3.3	58.3	52.6 ± 4.2	70.8 ± 2.8	61.0	54.8 ± 3.2	70.5 ± 3.0	**62.1**	53.6 ± 3.9	65.8 ± 3.2	59.4

TABLE 4.3 Sensitivity (%), Specificity (%) and Geometric Mean (%) of Hierarchical Feature Selection Methods with Naïve Bayes Classifier for *Drosophila melanogaster* Datasets

Drosophila melanogaster Datasets													
	Standard-NB			**HIP+NB**			**MR+NB**			**HIP–MR+NB**		
Thre.	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***
T3	73.8 ± 4.4	51.3 ± 10.5	61.5	75.0 ± 3.7	53.8 ± 10.0	**63.5**	70.0 ± 6.2	53.8 ± 10.2	61.4	81.3 ± 4.7	41.0 ± 7.6	57.7
T4	73.8 ± 4.4	48.7 ± 10.8	60.0	75.0 ± 2.6	51.3 ± 10.4	62.0	70.0 ± 3.8	59.0 ± 10.3	**64.3**	85.0 ± 4.5	43.6 ± 8.2	60.9
T5	70.0 ± 5.0	43.6 ± 4.5	55.2	76.3 ± 3.9	48.7 ± 3.2	**61.0**	70.0 ± 6.0	48.7 ± 4.1	58.4	80.0 ± 4.2	38.5 ± 4.0	55.5
T6	72.5 ± 4.5	43.6 ± 9.0	56.2	80.0 ± 3.3	46.2 ± 8.8	**60.8**	76.3 ± 4.4	43.6 ± 8.2	57.7	85.0 ± 3.1	35.9 ± 6.6	55.2
T7	76.3 ± 5.7	43.6 ± 6.3	57.7	80.0 ± 4.2	53.8 ± 8.8	**65.6**	76.3 ± 3.9	53.8 ± 8.0	64.1	83.8 ± 3.3	43.6 ± 6.3	60.4
T8	72.5 ± 4.5	43.6 ± 9.0	56.2	78.8 ± 5.6	48.7 ± 6.7	**61.9**	78.8 ± 5.6	43.6 ± 8.2	58.6	83.8 ± 5.3	38.5 ± 7.6	56.8
T9	75.0 ± 5.6	43.6 ± 10.2	57.2	82.5 ± 6.0	48.7 ± 10.4	**63.4**	75.0 ± 5.3	48.7 ± 10.4	60.4	77.5 ± 5.2	43.6 ± 8.7	58.1
T10	71.3 ± 4.6	41.0 ± 8.5	54.1	81.3 ± 3.8	43.6 ± 7.5	59.5	77.5 ± 4.5	46.2 ± 9.0	**59.8**	75.0 ± 4.6	46.2 ± 8.2	58.9
Ave.	73.2 ± 4.8	44.9 ± 8.6	57.3	78.6 ± 4.1	49.4 ± 8.2	**62.2**	74.2 ± 5.0	49.7 ± 8.6	60.6	81.4 ± 4.4	41.4 ± 7.2	57.9

TABLE 4.4 Sensitivity (%), Specificity (%) and Geometric Mean (%) of Hierarchical Feature Selection Methods with Naïve Bayes Classifier for *Mus musculus* Datasets

Mus musculus Datasets													
	Standard-NB			**HIP+NB**			**MR+NB**			**HIP–MR+NB**		
Thre.	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***
T3	79.4 ± 5.6	53.8 ± 8.5	65.4	82.5 ± 5.3	57.7 ± 9.7	**69.0**	63.5 ± 5.4	61.5 ± 8.9	62.5	96.8 ± 2.1	30.8 ± 7.4	54.6
T4	73.0 ± 6.4	46.2 ± 8.3	58.1	82.5 ± 5.5	57.7 ± 8.3	**69.0**	66.7 ± 6.2	61.5 ± 5.7	64.0	95.2 ± 3.5	34.6 ± 6.6	57.4
T5	81.0 ± 3.9	42.3 ± 8.3	58.5	82.5 ± 4.3	50.0 ± 9.9	**64.2**	71.4 ± 7.3	57.7 ± 10.9	**64.2**	93.7 ± 3.2	34.6 ± 6.8	56.9
T6	71.4 ± 2.8	46.2 ± 11.0	57.4	81.0 ± 4.2	46.2 ± 9.9	**61.2**	68.3 ± 5.9	53.8 ± 6.6	60.6	95.2 ± 2.4	34.6 ± 10.2	57.4
T7	74.6 ± 5.2	42.3 ± 6.2	56.2	84.1 ± 4.3	46.2 ± 8.2	**62.3**	69.8 ± 6.4	53.8 ± 9.6	61.3	90.5 ± 3.6	34.6 ± 8.4	56.0
T8	69.8 ± 5.7	42.3 ± 12.2	54.3	76.2 ± 6.5	50.0 ± 12.4	**61.7**	74.6 ± 4.5	50.0 ± 11.2	61.1	92.1 ± 2.6	34.6 ± 11.7	56.5
T9	74.6 ± 6.3	50.0 ± 14.2	61.1	81.0 ± 6.2	46.2 ± 11.1	61.2	73.0 ± 7.2	53.8 ± 12.2	**62.7**	92.1 ± 4.4	38.5 ± 12.7	59.5
T10	71.4 ± 5.6	57.7 ± 13.1	**64.2**	76.2 ± 6.0	42.3 ± 12.7	56.8	74.6 ± 5.5	46.2 ± 12.0	58.7	85.7 ± 5.6	30.8 ± 11.3	51.4
Ave.	74.4 ± 5.2	47.6 ± 10.2	59.4	80.8 ± 5.3	49.5 ± 10.3	**63.2**	70.2 ± 6.1	54.8 ± 9.6	61.9	92.7 ± 3.4	34.1 ± 9.4	56.2

TABLE 4.5 Sensitivity (%), Specificity (%) and Geometric Mean (%) of Hierarchical Feature Selection Methods with Naïve Bayes Classifier for *Saccharomyces cerevisiae* Datasets

Saccharomyces cerevisiae Datasets	**Standard-NB**			**HIP+NB**			**MR+NB**			**HIP–MR+NB**		
Thre.	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***	***Sen.***	***Spe.***	***GM***
T3	45.0 ± 7.3	82.2 ± 2.7	60.8	40.0 ± 8.5	92.3 ± 2.0	60.8	60.0 ± 8.5	80.3 ± 2.4	**69.4**	12.5 ± 7.7	98.1 ± 1.1	35.0
T4	55.0 ± 8.2	83.1 ± 3.4	**67.6**	47.5 ± 10.2	92.3 ± 2.2	66.2	52.5 ± 8.7	80.7 ± 2.9	65.1	20.0 ± 9.0	97.6 ± 1.3	44.2
T5	50.0 ± 6.5	82.0 ± 3.3	64.0	40.0 ± 7.6	90.3 ± 2.3	60.1	52.5 ± 6.9	80.1 ± 3.2	**64.8**	17.5 ± 6.5	97.6 ± 2.0	41.3
T6	45.0 ± 5.0	77.2 ± 2.7	58.9	40.0 ± 4.1	87.4 ± 2.7	59.1	55.0 ± 6.2	79.1 ± 3.5	**66.0**	17.5 ± 3.8	96.1 ± 1.9	41.0
T7	45.0 ± 8.2	78.6 ± 3.2	59.5	37.5 ± 5.6	91.3 ± 1.9	58.5	50.0 ± 6.5	85.0 ± 3.1	**65.2**	20.0 ± 7.3	97.1 ± 1.3	44.1
T8	45.0 ± 7.3	80.8 ± 2.1	60.3	35.0 ± 5.5	91.1 ± 1.7	56.5	45.0 ± 6.2	82.8 ± 3.8	**61.0**	15.0 ± 5.5	96.6 ± 1.2	38.1
T9	47.5 ± 5.8	78.6 ± 2.0	61.1	42.5 ± 9.9	88.1 ± 2.6	**61.2**	42.5 ± 6.5	79.1 ± 2.6	58.0	12.5 ± 5.6	96.0 ± 1.0	34.6
T10	52.5 ± 7.9	78.6 ± 1.9	**64.2**	37.5 ± 6.7	89.1 ± 1.5	57.8	37.5 ± 5.6	80.6 ± 1.2	55.0	20.0 ± 6.2	96.0 ± 1.5	43.8
Ave.	48.1 ± 7.0	80.1 ± 2.7	62.1	40.0 ± 7.3	90.2 ± 2.1	60.0	49.4 ± 6.9	81.0 ± 2.8	**63.1**	16.9 ± 6.5	96.9 ± 1.4	40.3

TABLE 4.6 Average Number of GO Terms Selected by Each Feature Selection Method for the 4 Model Organisms

	Caenorhabditis elegans			***Drosophila melanogaster***		
Thre.	***HIP+NB***	***MR+NB***	***HIP–MR+NB***	***HIP+NB***	***MR+NB***	***HIP–MR+NB***
T3	65.3	140.7	265.4	73.3	121.4	228.2
T4	58.6	113.2	223.6	65.2	101.5	190.7
T5	55.7	99.7	201.9	60.4	88.4	164.7
T6	52.4	87.7	182.2	51.9	73.7	139.7
T7	51.1	84.0	170.0	47.2	68.4	122.8
T8	49.4	73.0	152.6	44.4	62.1	108.9
T9	46.7	67.0	142.9	41.2	55.3	97.8
T10	45.5	63.3	135.9	38.8	47.6	87.1
Ave.	53.1	91.1	184.3	52.8	77.3	142.5
	Mus musculus			***Saccharomyces cerevisiae***		
Thre.	***HIP+NB***	***MR+NB***	***HIP–MR+NB***	***HIP+NB***	***MR+NB***	***HIP–MR+NB***
T3	120.6	178.5	330.3	54.3	99.6	218.7
T4	107.4	139.5	264.4	49.4	89.8	185.3
T5	93.1	114.8	215.9	44.5	73.2	151.3
T6	81.8	96.1	188.8	41.5	66.7	134.3
T7	71.8	78.3	160.9	37.3	57.2	117.1
T8	65.7	73.4	145.1	34.2	50.5	106.0
T9	61.0	68.0	133.7	33.2	46.0	98.5
T10	55.5	60.7	117.6	31.7	43.1	85.9
Ave.	82.1	101.2	194.6	40.8	65.8	137.1

The main reasons for the inferior performance of HIP–MR+NB seem to be that it tends to select a much larger number of GO term features, by comparison with HIP and MR (see Section 4.4), and such a larger feature subset contains some hierarchical redundancy among features (unlike the non-redundant features selected by HIP and MR), as explained earlier. As evidence for this, Table 4.6 shows the average number of features selected by each method for each model organism and each dataset version. Each value in the table is the mean number of selected features over the 10 cross-validation iterations. As shown in Table (4.6), the number of features selected by HIP–MR is always larger (and in most cases substantially larger) than the sum of the number of features selected by HIP and MR. Such larger feature subsets contain many hierarchically redundant features, reducing the predictive accuracy of Naïve Bayes with the HIP–MR method, since Naïve Bayes is sensitive to redundant features, as discussed in Chapter 2.

It is also worth observing the effect of different values of the GO term frequency threshold in the GMean value obtained by the different versions of Naïve Bayes in Tables 4.2 – 4.5. For standard-NB, the highest GMean value was obtained with the threshold 3 in the *Drosophila melanogaster* and *Mus musculus* datasets, threshold 4 in the *Saccharomyces cerevisiae* dataset, and threshold 6 in the *Caenorhabditis elegans* dataset. For HIP+NB, the highest GMean value was obtained with the threshold 3 or 4 in the *Mus musculus* dataset, threshold 4 in the *Saccharomyces cerevisiae* dataset, and threshold 7 in the *Caenorhabditis elegans* and *Drosophila melanogaster* datasets. For MR+NB, the highest GMean value was obtained with the threshold 3 in the *Saccharomyces cerevisiae* dataset, threshold 4 in the *Drosophila melanogaster* dataset, and threshold 5 in the *Caenorhabditis elegans* and *Mus musculus* datasets. For HIP–MR+NB, the highest GMean value was obtained with the threshold 4 in the *Drosophila melanogaster* and *Saccharomyces cerevisiae* datasets, threshold 6 in the *Caenorhabditis elegans* dataset, and threshold 9 in the *Mus musculus* dataset.

In summary, across the four versions of NB and the four model organisms, the most successful GO term frequency threshold value was 4, which led to the highest GMean value in 5.5 out of 16 cases – interpreting the tie between threshold values 3 and 4 for HIP+NB in the *Mus musculus* dataset as a count of "half-win" for each of those values. The second most successful GO term frequency threshold value was 3, which led to the highest GMean value in 3.5 out of 16 cases. That is, in 9

out of 16 cases the threshold value leading to the highest GMean value was either 3 or 4, which are the most inclusive threshold values – i.e. the values that lead to the largest number of GO term features used as input by the different versions of Naïve Bayes. Hence, broadly speaking, using lower, more inclusive threshold values seems more effective than higher, less inclusive threshold values, although the latter led to higher GMean values in several cases.

4.6.2 Discussion

We chose the combination of Friedman test and Holm *post-hoc* test as the statistical significance tests applied on the Geometric Mean values obtained for the 32 datasets used in our experiments (8 different GO term frequency thresholds times 4 model organisms). The Friedman test is a nonparametric test based on the rankings of each classifier's predictive performance on each dataset, which avoids the problems associated with the assumption of normal distribution made by the t-test and ANOVA [29, 58]. The Holm *post-hoc* method is used for coping with the multiple-comparison problem when using significance tests, by adjusting the p-values for individual pairwise comparisons. Demsăr [28] argues that in the case of multiple comparisons between one control classifier and other classifiers, the Holm *post-hoc* test is more powerful than the Nemenyi *post-hoc* test. We selected MR+NB as the control method, since it obtains the highest average GMean value (averaged over the 32 dataset versions) among the four methods being compared in Tables 4.2 – 4.5. Comparing the GMean values of MR+NB as the control method against the values of each of the other methods, at the significance level of 5%, there is no significant difference between the GMean values of MR+NB and HIP+NB; but MR+NB significantly outperforms both standard-NB and HIP–MR+NB.

Comparing the predictive accuracy of HIP+NB, MR+NB and HIP–MR+NB, it seems that hierarchical redundancy among the selected GO terms tends to decrease NB's predictive accuracy. As evidence for this, HIP–MR+NB, which selects a set of GO terms with some hierarchical redundancy, performed considerably worse than MR+NB and HIP+NB, which do not select hierarchically redundant features. Also, the core GO terms containing the complete hierarchical information in the GO DAG for a given instance seem valuable for prediction, since HIP+NB, which

selects such hierarchically non-redundant core GO terms regardless of relevance, performed about as well as MR+NB.

4.6.3 On the Statistical and Biological Relevance of a Number of Very Frequently Selected GO Terms

Recall that the proposed lazy hierarchical feature selection methods select a different set of features (GO terms) for each testing instance. Hence, when producing a ranking of GO terms in descending order of their usefulness, it is natural to calculate the ranking based on the number of instances where each GO term is selected to be used as input by Naïve Bayes. MR was overall the best feature selection method in terms of predictive performance in the experiments reported earlier. Hence, for the purpose of ranking the GO terms in decreasing order of frequency of selection, the ranking produced when using MR as the selection method is more appropriate, and this ranking criterion is used here.

For each model organism, we produced a ranking of all GO terms occurring in the dataset version with GO term frequency threshold 3 for that organism, since that dataset contains the largest number of GO terms. Note that the ranking criterion based on the frequency of selection when using the MR method does not directly take into account the statistical significance of selected GO terms. Some GO terms may be selected very often by MR due to their high relevance (predictive power), regardless of their statistical significance. Hence, to complement the ranking of GO terms based on their frequency of selection by MR, we also computed, for each GO term, its p-value associated with a statistical significance test, based on the following rationale [123].

If we had to predict the class of a gene based on a given GO term alone (without using any other feature), we would assign that gene to the class with the largest number of genes (instances) annotated with that GO term. We refer to that class as the class predicted by that GO term. The predictive accuracy associated with the use of that GO term as a predictor is the ratio of the number of instances that are annotated with that GO term and have the class predicted by the GO term divided by the number of instances that are annotated with that GO term.

To evaluate the statistical significance associated with a GO term used as a predictor, we use a significance test based on the binomial distribution, which has two parameters: n, the number of trials, and p, the probability of success in each trial. When applying the significance test, the assignment of the class predicted by the GO term to any given instance annotated with that term is regarded as a random trial with two possible results: success (the class predicted by the GO term equals the true class of that instance) or failure otherwise. The instances classified by the GO term are assumed to be independent from each other, and the number of trials n is the number of instances classified by the GO term – i.e. instances annotated with the GO term. Under the null hypothesis that the value "yes" ("1") of the GO term feature is irrelevant for predicting the class of an instance, the probability of observing a successful result is given by the relative frequency of the class predicted by the GO term in the dataset – i.e. the ratio of the number of instances of that class in the dataset divided by the total number of instances (of any class) in the dataset.

Hence, to set up a test of hypothesis for the statistical significance of the predictive power of a given GO term, we consider the observed number of instances that are correctly classified by the GO term, denoted k. That is, k is the number of instances that are annotated with the GO term and belong to the class predicted by the GO term. Let X be a random variable representing the number of successes in a binomial distribution with probability of success p and number of trials n. Under the null hypothesis that the GO term has no predictive power, for each model organism dataset version, the probability of observing exactly k successes, according to the binomial distribution, is given by Equation 4.3,

$$\mathbf{Pr}(X = k) = \mathbf{C}_k^n \mathbf{p}^k (1 - \mathbf{p})^{n-k}, \tag{4.3}$$

where $\mathbf{C}_k^n$ is the number of combinations of k elements out of n elements. Finally, for the test of hypothesis, we use Equation 4.3 to calculate the probability $\mathbf{Pr}(X \geq k)$. If the null hypothesis that the GO term has no predictive power can be rejected at the significant level of 5%, then the GO term's ability to predict its associated class can be considered as statistically significant.

We now discuss the relevance, to the biology of ageing, of 20 GO terms very

frequently selected as features by the MR method, among the set of terms whose predictive power was considered statistically significant (p-value < 0.05). The results are shown in Table 4.7, where the first three columns are self-explained. The fourth column shows the number (and %) of instances (in the dataset of the corresponding model organism) for which the GO term was selected by MR. The fifth column shows the rank of the GO term (the lower the rank, the better), among the set of GO terms whose p-value was deemed significant for the corresponding organism. The sixth and seventh columns show the p-value and the relevance value (computed by Equation 4.1) of the GO term. The eighth column shows the class predicted by each GO term. The following biological interpretation of the GO terms in Table 4.7 and their relevance to ageing was carried out by Dr. João Pedro de Magalhães, a biologist expert on ageing and co-author of our paper where such interpretation was reported [123].

Broadly speaking, the top ranking GO terms not only reflect our understanding of biological processes associated with ageing and life-extension in model organisms, but may help identify new putative associations suitable for further studies. As the organism in which single genes were initially associated with ageing, the roundworm *Caenorhabditis elegans* is arguably the best studied model in the context of ageing, with multiple pathways associated with the regulation of longevity [68]. It is the organism in which more gene manipulations have been shown to extend longevity [111] and unsurprisingly several top ranking GO categories in our results are known to impact on ageing. The top ranking term is "translation" with a strong association with anti-longevity. This is not surprising, since it is well-established that an inhibition of translation extends lifespan in *Caenorhabditis elegans* [68]. Other top categories like "autophagy", "apoptotic process", metabolism ("generation of precursor metabolites and energy") and maintenance of protein homeostasis ("response to topologically incorrect protein") have been linked to ageing [85]. Various top-ranked terms also relate to growth and development, which is not surprising given that developmental pathways in worms can significantly impact on ageing [26,68]. While all these results fit well with our current understanding of ageing, some categories may point towards novel mechanisms and warrant further investigation like "regulation of protein localization" and "transmembrane transport" associated, respectively, with pro- and anti-longevity.

TABLE 4.7 Information About 20 GO Terms Very Frequently Selected by the MR Method

Model Organism	GO Term ID	GO Term Name	Selection Frequency	Rank	P-Value	Relev.	Predicted Class
Caenorhabditis elegans	GO:0006412	translation	478 (100 %)	1	1.15 E-6	0.30	Anti
	GO:0006914	autophagy	478 (100 %)	3	1.57 E-3	0.50	Pro
	GO:0006915	apoptotic process	478 (100 %)	5	4.41 E-3	0.08	Anti
	GO:0006091	generation of precursor metabolites and energy	478 (100 %)	7	1.05 E-2	0.20	Anti
	GO:0032880	regulation of protein localization	478 (100 %)	8	1.82 E-2	0.30	Pro
	GO:0035966	response to topologically incorrect protein	478 (100 %)	9	2.41 E-2	0.23	Pro
	GO:0055085	transmembrane transport	435 (91.0 %)	24	5.26 E-5	0.21	Anti
Saccharomyces cerevisiae	GO:0001302	replicative cell aging	248 (100 %)	1	5.84 E-6	0.35	Pro
	GO:0000183	chromatin silencing at rDNA	248 (100 %)	2	5.67 E-4	0.73	Pro
	GO:0006302	double-strand break repair	248 (100 %)	3.5	7.71 E-3	0.45	Pro
	GO:0016265	death	244 (98.4 %)	6	1.48 E-2	0.53	Pro
	GO:0032200	telomere organization	243 (98.0 %)	7.5	2.95 E-3	0.64	Pro
Drosophila melanogaster	GO:0003006	developmental process involved in reproduction	119 (100 %)	1	3.48 E-3	0.30	Anti
	GO:0007600	sensory perception	119 (100 %)	2.5	1.15 E-2	0.55	Anti
	GO:0006629	lipid metabolic process	119 (100 %)	7	1.89 E-2	0.15	Pro
	GO:0055085	transmembrane transport	119 (100 %)	12	4.26 E-2	0.33	Anti
Mus musculus	GO:0040018	positive regulation of multicellular organism growth	89 (100 %)	2.5	7.28 E-3	0.65	Anti
	GO:0051093	negative regulation of developmental process	89 (100 %)	5	2.24 E-2	0.14	Pro
	GO:0010948	negative regulation of cell cycle process	78 (87.6 %)	19.5	2.24 E-2	0.14	Pro
	GO:0097190	apoptotic signaling pathway	75 (84.3 %)	21	4.04 E-2	0.10	Pro

A similar trend is observed in other model organisms. In yeast, which after worms is the model with most genes associated with ageing [111], top-ranked categories include "chromatin silencing at rDNA", "telomere organisation" and "double-strand break repair", all of which have been associated with longevity [85]; in addition to the expected "replicative cell ageing" and "death".

In flies, as in worms, some top terms are related to development, including the top category "developmental process involved in reproduction" associated with anti-longevity, and growth including cell division-related categories. Another top category associated with anti-longevity is "sensory perception", which fits well with recent results linking sensory perception, and olfactation in particular, to ageing [83]. Metabolism, with "lipid metabolic process" as the top category associated with pro-longevity, is in line with our understanding of life extension pathways mediated by diet, such as caloric restriction [97]. Intriguingly, "transmembrane transport" is, like in worms, also associated with anti-longevity, which merits further studies.

The top categories from mice partly reflect those found in lower model organisms, such as categories related to development and growth, like "positive regulation of multicellular organism growth" associated with anti-longevity and "negative regulation of developmental process" associated with pro-longevity. These results further emphasize the relationship between developmental processes and ageing, and further strengthen the idea that retarding development and growth can extend lifespan [26]. Also present in mice, as in invertebrates, are terms related to apoptosis ("apoptotic signaling pathway") and cell cycle ("negative regulation of cell cycle process"). Although this likely results from researcher biases, i.e. studying pathways in mice known to be associated with ageing in other model organisms, it highlights the evolutionary conservation of pathways associated with ageing [68].

4.7 Results Comparing Hierarchical and "Flat" Feature Selection Methods

4.7.1 The Feature Selection Methods Being Compared

We compared the hierarchical HIP and MR methods with three "flat" feature selection methods, i.e. Hybrid-lazy/eager-entropy-based feature selection [96], Hybrid-lazy/eager-relevance-based feature selection and Correlation-based Feature Selection (CFS). In these experiments we use only HIP and MR as hierarchical feature selection methods; we do not use the hybrid HIP–MR method because it performed clearly worse than HIP and MR in the experiments reported earlier. The main characteristics of the feature selection methods involved in the experiments are summarised in Table 4.8. The Hybrid-lazy/eager-entropy-based feature selection and Hybrid-lazy/eager-relevance-based feature selection methods follow the lazy learning scenario, i.e. conducting feature selection for each individual testing instance, although these two methods also have an "eager" learning component, as discussed next. In essence, these two methods measure the quality of each feature, and then produce a ranking of all the features based on that measure and select the top k features in that ranking.

The difference between those two methods is the feature quality measure: one uses entropy, as shown in Equation 4.4 [96]. This method calculates two versions of a feature's entropy: in the lazy version, the entropy is calculated using only the training instances with the value v_j ("1" or "0") of the feature A_j observed in the current testing instance being classified; whilst in the eager version, the entropy is calculated using all training instances, regardless of the value v_j observed in the current testing instance. Then the method chooses the smaller of these two entropy values as the feature's quality measure.

$$Ent(A_j, v_j) = min(Ent(A_j, v_j), Ent(A_j)) \tag{4.4}$$

The other method uses the relevance measure given by Equation 4.1, which follows the eager scenario, i.e. calculating the relevance value of each feature using

TABLE 4.8 Summary of Characteristics of Feature Selection Methods Working with Naïve Bayes

Feature Selection Method	***Learning Approach***	***Annotations***
No Feature Selection	Eager	Performs as Standard NB
HIP	Lazy	
MR	Lazy	
Entropy-based(HIP-k)	Hybrid	Select the same No. of features selected by HIP
Entropy-based(MR-k)	Lazy/Eager	Select the same No. of features selected by MR
Relevance-based(HIP-k)	Hybrid	Select the same No. of features selected by HIP
Relevance-based(MR-k)	Lazy/Eager	Select the same No. of features selected by MR
CFS	Eager	

all training instances. This is a hybrid lazy/eager method because the measure of relevance is calculated using the whole training dataset in an "eager" approach, but it selects the top-k ranked features for each testing instance, in a "lazy" approach.

For both methods, the parameter k, representing the number of features selected for each instance, equals to the number of features selected by the HIP or MR method respectively. That is, for each testing instance, the Hybrid-lazy/eager-entropy-based feature selection method and the Hybrid-lazy/eager-relevance-based feature selection method will select the same number of features selected by HIP or MR. This adds a lazy criterion to both these methods, since HIP and MR are lazy methods.

In contrast, CFS is an eager feature selection method that selects a single feature subset for all testing instances. CFS does not require a parameter specifying the number of features to be selected. It tries to select a subset of features that

have a high correlation with the class variable and have low redundancy among the features in the selected subset [48].

4.7.2 Dataset Creation

We created 28 datasets following essentially the methodology for creating datasets explained in Section 4.5.1. For each model organism, we created 7 datasets, with all possible subsets of the three GO term types, i.e. one dataset for each type of GO term (BP, MF, CC), one dataset for each pair of GO term types (BP and MF, BP and CC, MF and CC), and one dataset with all three GO term types (BP, MF and CC). Note that, in the case of generating datasets that are composed by different types of GO terms, their corresponding DAGs have sets of nodes that do not intersect each other. For example, when generating datasets consisting of the BP and MF types of GO terms, the corresponding BP and MF DAGs are separated, with no intersection. This also means that the hierarchical feature selection methods conduct the feature selection based on each individual DAG separately. In addition, the root terms for the DAG of biological process (GO:0008150); molecular function (GO:0003674), and cellular component (GO:0005575) terms are merely used for generating the datasets, but not included in the corresponding datasets used for experiments, due to their uselessness in terms of predictive power. In terms of the threshold for the minimum number of occurrences of a GO term, according to the discussion in Section 4.6.1, the value of this threshold is defined as 3, which retains more biological information than higher thresholds while still leading to high predictive accuracy. The detailed information about the created datasets is shown in Table 4.9, where the numbers of features, edges, instances and the degree of class imbalance are reported. The degree of class imbalance is calculated by Equation 4.5, where the degree equals to the complement of the ratio of the number of instances belonging to the minority class (**No**.($Minor$)) over the number of instances belonging to the majority class (**No**.($Major$)).

$$\textbf{Degree} = 1 - \frac{\textbf{No}.(Minor)}{\textbf{No}.(Major)} \tag{4.5}$$

TABLE 4.9 Main Characteristics of the Created Datasets with GO Term Frequency Threshold = 3

Caenorhabditis elegans							
Property	**BP**	**MF**	**CC**	**BP+MF**	**BP+CC**	**MF+CC**	**BP+MF+CC**
No. of Features	830	218	143	1048	973	361	1191
No. of Edges	1437	259	217	1696	1654	476	1913
No. of Instances	528	279	254	553	557	432	572
No. (%) of Pro-	209	121	98	213	213	170	215
Longevity Instances	39.6%	43.4%	38.6%	38.5%	38.2%	39.4%	37.6%
No. (%) of Anti-	319	158	156	340	344	262	357
Longevity Instances	60.4%	56.6%	61.4%	61.5%	61.8%	60.6%	62.4%
Degree of Class Imbalance	0.345	0.234	0.372	0.374	0.381	0.351	0.398
Drosophila melanogaster							
Property	**BP**	**MF**	**CC**	**BP+MF**	**BP+CC**	**MF+CC**	**BP+MF+CC**
No. of Features	698	130	75	828	773	205	903
No. of Edges	1190	151	101	1341	1291	252	1442
No. of Instances	127	102	90	130	128	123	130
No. (%) of Pro-	91	68	62	92	91	85	92
Longevity Instances	71.7%	66.7%	68.9%	70.8%	71.1%	69.1%	70.8%
No. (%) of Anti-	36	34	28	38	37	38	38
Longevity Instances	28.3%	33.3%	31.1%	29.2%	28.9%	30.9%	29.2%
Degree of Class Imbalance	0.604	0.500	0.548	0.587	0.593	0.553	0.587
Mus musculus							
Property	**BP**	**MF**	**CC**	**BP+MF**	**BP+CC**	**MF+CC**	**BP+MF+CC**
No. of Features	1039	182	117	1221	1156	299	1338
No. of Edges	1836	205	160	2041	1996	365	2201
No. of Instances	102	98	100	102	102	102	102
No. (%) of Pro-	68	65	66	68	68	68	68
Longevity Instances	66.7%	66.3%	66.0%	66.7%	66.7%	66.7%	66.7%
No. (%) of Anti-	34	33	34	34	34	34	34
Longevity Instances	33.3%	33.7%	34.0%	33.3%	33.3%	33.3%	33.3%
Degree of Class Imbalance	0.500	0.492	0.485	0.500	0.500	0.500	0.500
Saccharomyces cerevisiae							
Property	**BP**	**MF**	**CC**	**BP+MF**	**BP+CC**	**MF+CC**	**BP+MF+CC**
No. of Features	679	175	107	854	786	282	961
No. of Edges	1223	209	168	1432	1391	377	1600
No. of Instances	215	157	147	222	234	226	238
No. (%) of Pro-	30	26	24	30	30	29	30
Longevity Instances	14.0%	16.6%	16.3%	13.5%	12.8%	12.8%	12.6%
No. (%) of Anti-	185	131	123	192	204	197	208
Longevity Instances	86.0%	83.4%	83.7%	86.5%	87.2%	87.2%	87.4%
Degree of Class Imbalance	0.838	0.802	0.805	0.844	0.853	0.853	0.856

4.7.3 Experimental Results Comparing HIP and MR with Other Feature Selection Methods

Tables 4.10 – 4.17 report the results for the hierarchical and "flat" feature selection methods working with the Naïve Bayes classifier. In these tables, the numbers after the symbol "±" denote standard errors. We also show, in Figure 4.3 and 4.4, the average ranks based on the GMean values for different feature selection methods working with Naïve Bayes.

Tables 4.10 – 4.13 compare the predictive accuracies obtained by Naïve Bayes when using 4 different feature selection methods in a pre-processing phase: one of the hierarchical feature selection methods, namely HIP; two hybrid lazy/eager "flat" (non-hierarchical) feature selection methods, namely Hybrid-lazy/eager-entropy-based (selecting the same number of k features as HIP) (Ent_{HIP-k}) and Hybrid-lazy/eager-relevance-based (selecting the same number of k features as HIP) (Rele_{HIP-k}); and one eager "flat" feature selection method, namely CFS. The tables also report results for Naïve Bayes (NB) without using any feature selection method, as a natural baseline.

In details, for the results about *Caenorhabditis elegans* in Table 4.10, the values of specificity are greater than the values of sensitivity obtained by all algorithms, since in this dataset sensitivity is the predictive accuracy for the minority class, whose prediction is in general more difficult, due to less data to support such prediction. HIP+NB obtains the highest GMean value 6 out of 7 times, while Rele_{HIP-k}+NB obtains one time the highest GMean value. In Table 4.11, for the results about *Drosophila melanogaster*, the values of sensitivity are greater than the values of specificity obtained by all algorithms, since in this dataset specificity represents the predictive accuracy for the minority class. HIP+NB obtains 5 out of 7 times the highest GMean value, and Rele_{HIP-k}+NB and NB without feature selection obtains each one time the highest GMean value. Analogous to Table 4.11, overall, the values of sensitivity are greater than the values of specificity obtained by all algorithms shown in Table 4.12. HIP+NB obtains 5 out of 7 times the highest GMean value, while NB without feature selection obtains two times the highest GMean value. For the results about *Saccharomyces cerevisiae* in Table 4.13, the values of specificity are greater than the values of sensitivity.

HIP+NB obtains 5 out of 7 times the highest GMean value, while NB without feature selection obtains two times the highest GMean value.

The HIP+NB method obtains the best results with the average rank of 1.43, while the second best rank (2.38) was obtained by NB without feature selection. The average rank for CFS+NB is 3.18, and the average rank for Rele$_{HIP-k}$+NB is 3.23, whereas Ent$_{HIP-k}$+NB obtained the worst average rank (4.79) in terms of GMean value. It is obvious that the HIP method performs best when it works with Naïve Bayes, since it ranks in the first position in 21 out of 28 datasets, as indicated by the boldfaced GMean values in Tables 4.10 – 4.13.

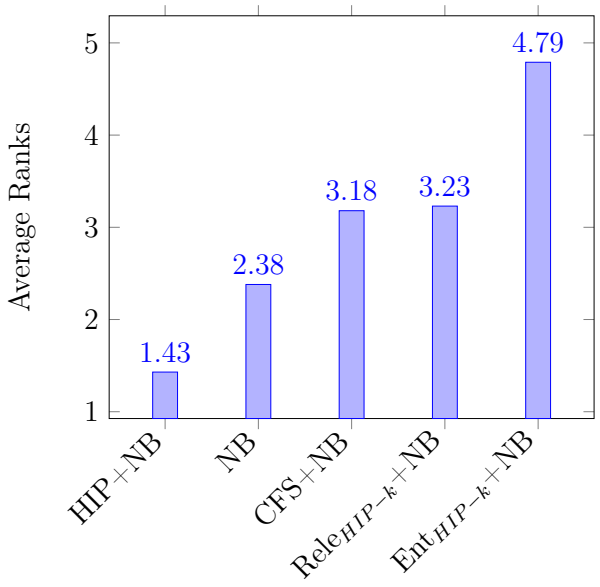


FIGURE 4.3 Summary of Methods' Average Ranks from Tables 4.10 – 4.13

Tables 4.14 – 4.17, which are analogous to Tables 4.10 – 4.13, compare the predictive accuracy of Naïve Bayes using the hierarchical feature selection method MR with the accuracies of Naïve Bayes using 3 "flat" feature selection methods, namely Ent$_{MR-k}$, Rele$_{MR-k}$ and CFS, and NB without feature selection.

In details, for the results about *Caenorhabditis elegans* in Table 4.14, the values of specificity are greater than the values of sensitivity obtained by all algorithms. MR+NB obtains the highest GMean value 3 out of 7 times, while Rele$_{MR-k}$, Ent$_{MR-k}$, CFS and NB without feature selection obtain each one time the highest GMean value. In Table 4.15, for the results about *Drosophila melanogaster*, the values of sensitivity are greater than the values of specificity obtained by all algorithms. MR+NB obtains 5 out of 7 times the highest GMean value, NB without

feature selection obtains two times the highest GMean value, and and Rele$_{MR-k}$ obtains one time the highest value (one draw with MR+NB). For the results about *Mus musculus* in Table 4.16, the values of sensitivity are greater than the values of specificity obtained by all algorithms. NB without feature selection obtains 3 out of 7 times the highest GMean value, while MR+NB and CFS+NB obtain each two times the highest GMean value. For the results about *Saccharomyces cerevisiae* in Table 4.17, the values of specificity are greater than the values of sensitivity. NB without feature selection obtains 5 out of 7 times the highest GMean value, while Rele$_{MR-k}$+NB obtains two times of the highest GMean value.

NB without using any feature selection method has the best average rank of 2.18 over all datasets, while the average rank for MR+NB is 2.39, which is better than the average rank obtained by Rele$_{MR-k}$+NB (2.57), CFS+NB (3.09) and Ent$_{MR-k}$+NB (4.77). MR+NB obtained the highest GMean in 10 out of 28 datasets, as indicated by the boldfaced GMean values in Tables 4.14 – 4.17; whilst NB without feature selection did slightly better, with the highest GMean in 11 out of 28 datasets.

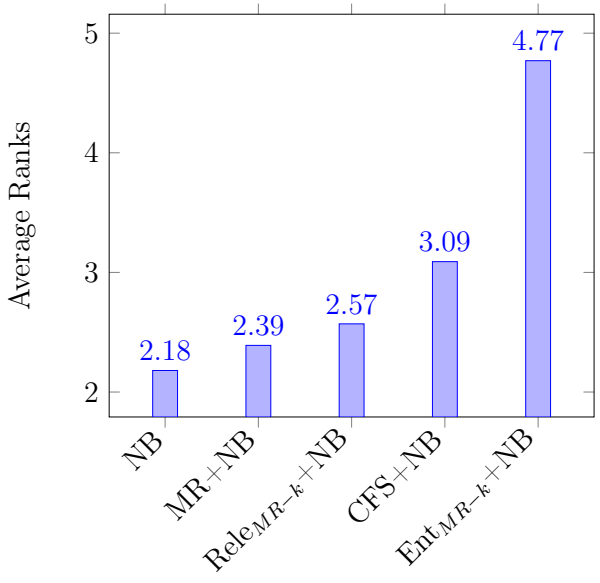


FIGURE 4.4 Summary of Methods' Ranks from Tables 4.14 – 4.17

TABLE 4.10 Predictive Accuracy for Naïve Bayes with the Hierarchical HIP Method and Baseline "Flat" Feature Selection Methods for *Caenorhabditis elegans* Datasets

Feature	**NB without**			**Lazy HIP + NB**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + NB**		
Type	**Feature Selection**						**Ent$_{HIP-k}$ + NB**			**Rele$_{HIP-k}$ + NB**					
	Caenorhabditis elegans Datasets														
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	50.2±3.6	69.0±2.6	58.9	54.1±3.4	75.5±2.8	**63.9**	34.4±3.0	84.0±2.0	53.8	35.9±2.8	81.2±2.6	54.0	41.1±3.3	83.7±2.6	58.7
MF	57.9±4.1	46.2±5.5	51.7	45.5±4.7	51.9±5.1	48.6	36.4±2.8	65.2±4.4	48.7	66.9±7.7	43.7±5.8	**54.1**	58.7±6.8	46.8±5.5	52.4
CC	43.9±5.7	70.5±3.4	55.6	58.2±4.9	60.9±4.0	**59.5**	20.4±3.0	83.3±2.6	41.2	25.5±4.2	79.5±3.4	45.0	35.7±4.3	74.4±3.9	51.5
BP+MF	54.0±1.8	70.3±3.0	61.6	53.5±3.6	76.2±1.9	**63.8**	30.5±1.5	85.6±1.3	51.1	38.5±3.8	79.4±2.3	55.3	50.2±3.5	77.1±2.4	62.2
BP+CC	52.6±3.9	68.3±2.6	59.9	57.7±3.7	73.0±2.6	**64.9**	27.7±2.7	85.5±2.4	48.7	37.6±2.7	81.1±2.1	55.2	44.6±3.7	77.0±2.2	58.6
MF+CC	51.2±2.8	64.1±4.3	57.3	54.7±3.3	66.0±4.1	**60.1**	39.4±4.2	80.5±3.5	56.3	37.6±3.3	76.3±3.5	53.6	47.1±3.9	72.1±3.8	58.3
BP+MF+CC	52.1+4.4	70.0+2.3	60.4	55.3+3.6	71.7+2.7	**63.0**	29.3+3.4	84.9+1.8	49.9	45.6+3.9	80.1+2.0	60.4	51.6+4.4	74.8+2.1	62.1

TABLE 4.11 Predictive Accuracy for Naïve Bayes with the Hierarchical HIP Method and Baseline "Flat" Feature Selection Methods for *Drosophila melanogaster* Datasets

Feature		***Drosophila melanogaster* Datasets**													
	NB without		**Lazy HIP + NB**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + NB**		
Type	**Feature Selection**					**Ent**$_{HIP-k}$ **+ NB**			**Rele**$_{HIP-k}$ **+ NB**					
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	74.7±3.5	36.1±9.5	51.9	73.6±4.1	44.4±9.0	57.2	93.4±2.5	2.8±2.5	16.2	76.9±3.2	47.2±8.2	**60.2**	76.9±4.7	27.8±7.4	46.2
MF	82.4±4.6	35.3±8.6	53.9	69.1±6.1	52.9±7.3	**60.5**	97.1±2.3	32.4±6.3	56.1	92.6±3.4	32.4±9.5	54.8	86.8±4.0	35.3±7.2	55.4
CC	87.1±4.1	50.0±10.2	**66.0**	80.6±6.5	46.4±11.4	61.2	91.9±2.7	25.0±7.1	47.9	85.5±5.2	39.3±8.7	58.0	87.1±3.3	39.3±10.0	58.5
BP+MF	77.2±3.9	50.0±10.2	62.1	72.8±5.6	57.9±9.3	**64.9**	95.7±2.5	15.8±7.6	38.9	84.8±3.0	44.7±10.8	61.6	85.9±3.7	31.6±7.5	52.1
BP+CC	76.9±5.1	48.6±9.8	61.1	73.6±4.9	64.9±8.3	**69.1**	91.2±3.5	2.7±2.5	15.7	78.0±4.0	40.5±10.2	56.2	82.4±3.7	43.2±10.9	59.7
MF+CC	89.4±3.2	57.9±5.3	71.9	82.4±6.1	63.2±6.7	**72.2**	95.3±2.5	34.2±5.5	57.1	91.8±3.1	47.4±4.5	66.0	91.8±3.4	42.1±8.4	62.2
BP+MF+CC	81.5±5.3	55.3±8.2	67.1	76.1±4.9	68.4±5.3	**72.1**	96.7±1.7	21.1±8.7	45.2	87.0±3.2	50.0±8.3	66.0	90.2±3.1	47.4±8.7	65.4

TABLE 4.12 Predictive Accuracy for Naïve Bayes with the Hierarchical HIP Method and Baseline "Flat" Feature Selection Methods for *Mus musculus* Datasets

Feature	**NB without**			**Lazy HIP + NB**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + NB**		
Type	**Feature Selection**						**Ent**$_{HIP-k}$ **+ NB**			**Rele**$_{HIP-k}$ **+ NB**					
	Mus musculus Datasets														
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	82.4±4.7	44.1±5.9	60.3	72.1±4.8	70.6±5.1	**71.3**	95.6±2.2	29.4±4.1	53.0	91.2±3.2	44.1±7.0	63.4	83.8±4.0	38.2±5.6	56.6
MF	69.2±7.4	48.5±11.2	57.9	78.5±4.4	45.5±12.2	**59.8**	87.7±3.0	30.3±10.8	51.5	84.6±3.7	36.4±11.9	55.5	80.0±5.2	36.4±10.5	54.0
CC	75.8+2.3	52.9+10.0	**63.3**	80.3+3.0	47.1+11.2	61.5	81.8+3.3	32.4+11.7	51.5	75.8+3.2	41.2+11.9	55.9	71.2+3.0	35.3+11.2	50.1
BP+MF	83.8±3.4	44.1±7.0	60.8	70.6±4.8	70.6±8.1	**70.6**	94.1±2.3	32.4±6.4	55.2	86.8±4.5	44.1±7.2	61.9	88.2±4.2	41.2±8.0	60.3
BP+CC	79.4±6.1	50.0±8.4	63.0	66.2±5.0	73.5±9.3	**69.8**	97.1±1.9	32.4±8.9	56.1	88.2±4.7	38.2±10.3	58.0	83.8±5.0	50.0±11.3	64.7
MF+CC	75.0±5.0	64.7±12.5	**69.7**	79.4±4.2	58.8±11.8	68.3	91.2±3.3	32.4±8.9	54.4	83.8±5.0	47.1±10.5	62.8	77.9±4.8	47.1±10.9	60.6
BP+MF+CC	82.4±4.2	47.1±9.3	62.3	73.5±5.1	73.5±9.8	**73.5**	92.6±4.4	35.3±9.4	57.2	85.3±4.3	41.2±9.1	59.3	83.8±3.3	52.9±6.8	66.6

TABLE 4.13 Predictive Accuracy for Naïve Bayes with the Hierarchical HIP Method and Baseline "Flat" Feature Selection Methods for *Saccharomyces cerevisiae* Datasets

Feature	***Saccharomyces cerevisiae Datasets***	**NB without**			**Lazy HIP + NB**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + NB**		
Type		**Feature Selection**						**Ent$_{HIP-k}$ + NB**			**Rele$_{HIP-k}$ + NB**					
		Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP		40.0±8.3	84.9±3.5	58.3	63.3±6.0	78.4±3.1	**70.4**	3.3±3.3	100.0±0.0	18.2	40.0±6.7	84.3±3.7	58.1	20.0±5.4	91.4±2.6	42.8
MF		11.5±6.1	81.7±4.8	**30.7**	5.0±5.0	83.2±3.4	20.4	0.0±0.0	98.5±1.0	0.0	7.7±4.4	90.8±3.3	26.4	3.8±1.2	92.4±1.8	18.7
CC		25.0±7.1	86.2±3.0	46.4	29.2±10.2	82.9±4.2	**49.2**	16.7±7.0	95.1±1.7	39.9	20.8±6.9	87.8±3.1	42.7	20.8±7.5	94.3±1.7	44.3
BP+MF		33.3±11.1	85.4±1.7	53.3	76.7±7.1	74.0±3.3	**75.3**	0.0±0.0	100.0±0.0	0.0	50.0±7.5	85.9±1.9	65.5	33.3±9.9	90.6±1.5	54.9
BP+CC		53.3±8.9	85.8±3.0	67.6	70.0±7.8	79.4±3.2	**74.6**	0.0±0.0	100.0±0.0	0.0	36.7±10.5	85.3±2.5	56.0	40.0±8.3	91.2±1.8	60.4
MF+CC		34.5±10.5	87.3±2.1	**54.9**	31.0±8.0	82.2±3.5	50.5	6.9±5.7	95.9±1.3	25.7	24.1±9.7	89.8±1.7	46.5	13.8±6.3	91.9±1.9	35.6
BP+MF+CC		36.7±9.2	85.6±2.7	56.0	70.0±10.5	75.0±2.6	**72.5**	0.0±0.0	100.0±0.0	0.0	30.0±9.2	87.0±1.7	51.1	36.7±10.5	92.8±1.9	58.4

TABLE 4.14 Predictive Accuracy for Naïve Bayes with the Hierarchical MR Method and Baseline "Flat" Feature Selection Methods for *Caenorhabditis elegans* Datasets

Feature	**NB without**			**Lazy MR + NB**			**Lazy/Eager**			**Lazy/Eager**			***Caenorhabditis elegans Datasets***			**Eager CFS + NB**		
Type	**Feature Selection**						**Ent**$_{MR-k}$ **+ NB**			**Rele**$_{MR-k}$ **+ NB**								
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*		*Sen.*	*Spe.*	*GM*		
BP	50.2±3.6	69.0±2.6	58.9	51.2±3.5	75.5±2.6	**62.2**	32.1±1.8	83.7±2.6	51.8	46.4±3.0	73.4±2.4	58.4		41.1±3.3	83.7±2.6	58.7		
MF	57.9±4.1	46.2±5.5	51.7	38.8±2.9	63.3±3.8	49.6	47.9±3.4	58.2±5.2	**52.8**	76.0±6.9	35.4±6.3	51.9		58.7±6.8	46.8±5.5	52.4		
CC	43.9±5.7	70.5±3.4	**55.6**	42.9±4.0	71.2±3.0	55.3	22.4±3.2	80.8±3.2	42.5	37.8±5.3	73.1±3.2	52.6		35.7±4.3	74.4±3.9	51.5		
BP+MF	54.0±1.8	70.3±3.0	61.6	62.9±3.5	73.2±1.8	**67.9**	31.5±1.8	80.9±2.0	50.5	57.3±4.4	71.5±2.1	64.0		50.2±3.5	77.1±2.4	62.2		
BP+CC	52.6+3.9	68.3±2.6	59.9	55.4±2.8	73.8±2.2	**63.9**	32.9±2.7	81.7±2.1	51.8	50.2±3.1	75.6±2.1	61.6		44.6±3.7	77.0±2.2	58.6		
MF+CC	51.2±2.8	64.1±4.3	57.3	47.6±3.6	68.3±4.2	57.0	39.4±4.4	77.5±4.1	55.3	48.2±2.4	70.2±2.9	58.2		47.1±3.9	72.1±3.8	**58.3**		
BP+MF+CC	52.1+4.4	70.0+2.3	60.4	55.8+3.6	70.6+2.4	62.8	31.6+3.9	81.5+2.2	50.7	54.9+3.3	74.8+2.5	**64.1**		51.6+4.4	74.8+2.1	62.1		

TABLE 4.15 Predictive Accuracy for Naïve Bayes with the Hierarchical MR Method and Baseline "Flat" Feature Selection Methods for *Drosophila melanogaster* Datasets

Feature	**NB without**			**Lazy MR + NB**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + NB**		
Type	**Feature Selection**						**Ent**$_{MR-k}$ **+ NB**			**Rele**$_{MR-k}$ **+ NB**					
	Drosophila melanogaster Datasets														
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	74.7±3.5	36.1±9.5	51.9	79.1±4.1	38.9±11.0	**55.5**	94.5±2.5	8.3±4.3	28.0	79.1±2.4	38.9±10.9	**55.5**	76.9±4.7	27.8±7.4	46.2
MF	82.4±4.6	35.3±8.6	53.9	80.9±4.2	44.1±7.6	**59.7**	95.6±2.5	29.4±7.2	53.0	89.7±3.8	35.3±10.1	56.3	86.8±4.0	35.3±7.2	55.4
CC	87.1±4.1	50.0±10.2	66.0	83.9±5.6	53.6±8.7	**67.1**	95.2±2.4	21.4±7.4	45.1	87.1±4.1	39.3±8.7	58.5	87.1±3.3	39.3±10.0	58.5
BP+MF	77.2±3.9	50.0±10.2	**62.1**	79.3±4.3	44.7±8.2	59.5	94.6±3.4	10.5±4.1	31.5	81.5±3.4	44.7±11.5	60.4	85.9±3.7	31.6±7.5	52.1
BP+CC	76.9±5.1	48.6±9.8	61.1	80.2±4.3	56.8±11.2	**67.5**	93.4±2.8	2.7±2.5	15.9	81.3±4.0	40.5±9.0	57.4	82.4±3.7	43.2±10.9	59.7
MF+CC	89.4±3.2	57.9±5.3	**71.9**	83.5±4.4	57.9±7.5	69.5	96.5±1.8	34.2±6.7	57.4	92.9±1.9	44.7±5.0	64.4	91.8±3.4	42.1±8.4	62.2
BP+MF+CC	81.5±5.3	55.3±8.2	67.1	77.2±4.5	63.2±7.7	**69.9**	95.7±1.8	13.2±5.5	35.5	85.9±3.7	50.0±7.5	65.5	90.2±3.1	47.4±8.7	65.4

TABLE 4.16 Predictive Accuracy for Naïve Bayes with the Hierarchical MR Method and Baseline "Flat" Feature Selection Methods for *Mus musculus* Datasets

Feature		**NB without**			**Lazy MR + NB**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + NB**		
Type	***Mus musculus Datasets***	**Feature Selection**						**Ent_{MR-k} + NB**			**Rele_{MR-k} + NB**					
		Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP		82.4±4.7	44.1±5.9	60.3	80.9±5.2	50.0±7.9	**63.6**	97.1±1.9	32.4±4.5	56.1	89.7±3.7	44.1±7.0	62.9	83.8±4.0	38.2±5.6	56.6
MF		69.2±7.4	48.5±11.2	**57.9**	83.1±4.1	39.4±10.7	57.2	87.7±3.0	24.2±9.0	46.1	84.6±3.7	39.4±13.0	57.7	80.0±5.2	36.4±10.5	54.0
CC		75.8+2.3	52.9+10.0	**63.3**	81.8+3.6	41.2+11.9	58.1	81.8+3.3	29.4+11.0	49.0	75.8+2.3	44.1+11.1	57.8	71.2+3.0	35.3+11.2	50.1
BP+MF		83.8±3.4	44.1±7.0	60.8	82.4±4.2	50.0±10.2	**64.2**	97.1±1.9	35.3±7.3	58.5	86.8±4.0	38.2±6.2	57.6	88.2±4.2	41.2±8.0	60.3
BP+CC		79.4±6.1	50.0±8.4	63.0	73.5±5.1	52.9±9.6	62.4	95.6±3.0	29.4±8.7	53.0	88.2±5.1	47.1±9.7	64.5	83.8±5.0	50.0±11.3	**64.7**
MF+CC		75.0±5.0	64.7±12.5	**69.7**	83.8±5.0	55.9±13.3	68.4	91.2±3.3	29.4±8.1	51.8	80.9±5.2	52.9±11.3	65.4	77.9±4.8	47.1±10.9	60.6
BP+MF+CC		82.4±4.2	47.1±9.3	62.3	85.3±4.3	50.0±6.9	65.3	94.1±3.2	35.3±9.4	57.6	85.3±4.3	44.1±8.9	61.3	83.8±3.3	52.9±6.8	**66.6**

TABLE 4.17 Predictive Accuracy for Naïve Bayes with the Hierarchical MR Method and Baseline "Flat" Feature Selection Methods for *Saccharomyces cerevisiae* Datasets

Feature	**NB without**			**Lazy MR + NB**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + NB**		
Type	**Feature Selection**						**Ent**$_{MR-k}$ **+ NB**			**Rele**$_{MR-k}$ **+ NB**					
	Saccharomyces cerevisiae Datasets														
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	40.0±8.3	84.9±3.5	**58.3**	33.3±8.6	85.9±2.9	53.5	0.0±0.0	98.4±1.2	0.0	36.7±10.5	86.5±2.8	56.3	20.0±5.4	91.4±2.6	42.8
MF	11.5±6.1	81.7±4.8	**30.7**	0.0±0.0	93.9±2.4	0.0	0.0±0.0	98.5±1.0	0.0	3.8±3.3	90.1±2.9	18.5	3.8±1.2	92.4±1.8	18.7
CC	25.0±7.1	86.2±3.0	**46.4**	20.8±6.9	91.9±2.7	43.7	16.7±7.0	95.1±1.7	39.9	20.8±7.5	87.8±2.2	42.7	20.8±7.5	94.3±1.7	44.3
BP+MF	33.3±11.1	85.4±1.7	53.3	23.3±5.1	89.1±2.5	45.6	0.0±0.0	97.9±0.8	0.0	36.7±6.0	84.9±1.5	**55.8**	33.3±9.9	90.6±1.5	54.9
BP+CC	53.3±8.9	85.8±3.0	**67.6**	40.0±8.3	84.8±2.7	58.2	10.0±5.1	99.0±0.7	31.5	43.3±8.7	88.7±1.6	62.0	40.0±8.3	91.2±1.8	60.4
MF+CC	34.5±10.5	87.3±2.1	**54.9**	17.2±6.3	89.8±2.3	39.3	13.8±6.3	94.9±1.3	36.2	20.7±10.0	88.8±1.6	42.9	13.8±6.3	91.9±1.9	35.6
BP+MF+CC	36.7±9.2	85.6±2.7	56.0	30.0±9.2	86.5±2.6	50.9	0.0±0.0	99.5±0.5	0.0	43.3±11.2	90.9±1.4	**62.7**	36.7±10.5	92.8±1.9	58.4

4.7.4 Discussion

4.7.4.1 Statistical Analysis of GMean Value Differences between HIP or MR and Other Feature Selection Methods

We adopted the Friedman test and Holm *post-hoc* method to conduct the statistical significance test on the differences between the GMean values of feature selection methods working with Naïve Bayes. The results of the statistical significance tests are shown in Table 4.18, where columns 2 and 6 present the average ranks of different feature selection methods; columns 3 and 7 present the corresponding p-values, and columns 4 and 8 present the adjusted significance level according to Holm *post-hoc* method. The boldfaced p-values indicate that the corresponding results are significant at the α=0.05 significance level, which occurs when the p-value is smaller than the "Adjusted α".

As shown on the left 4 columns of Table 4.18, HIP (the control method) is compared with other feature selection methods. The outcome shows that HIP significantly improves the performance of NB without feature selection, and significantly outperforms Relevance-based (Rele_{HIP-k}), Entropy-based (Ent_{HIP-k}) and the CFS feature selection methods, when working with NB.

On the right 4 columns of Table 4.18, the MR method is used as the control method, although NB without feature selection obtained the best rank since in the context of this thesis it is more important to evaluate the predictive performance of the new MR method than the performance of NB without feature selection. When those feature selection methods work with NB, MR significantly outperforms the Entropy-based (Ent_{MR-k}) feature selection method, but it shows no significant difference to other feature selection methods.

TABLE 4.18 Statistical Significance Test Results of the Algorithms' GMean Values According to the Non-Parametric Friedman Test with the Holm *Post-Hoc* Test at the $\alpha = 0.05$ Significance Level

FS Method	**Ave. Rank**	**P-Value**	**Adjusted α**	**FS Method**	**Ave. Rank**	**P-Value**	**Adjusted α**
HIP (ctrl.)	1.43	–	–	MR (ctrl.)	2.39	–	–
No FS	2.38	**2.47 E-02**	0.0500	No FS	2.18	6.19 E-01	0.0500
CFS	3.18	**3.52 E-05**	0.0250	Rele_{MR-k}	2.57	6.70 E-01	0.0250
Rele_{HIP-k}	3.23	**2.09 E-05**	0.0167	CFS	3.09	9.76 E-02	0.0167
Ent_{HIP-k}	4.79	**1.97 E-15**	0.0125	Ent_{MR-k}	4.77	**1.78 E-08**	0.0125

4.7.4.2 Analysis of the Correlation between GMean Values and Degrees of Class Imbalance for the HIP and MR Methods

As shown in Figure 4.5, the values of the degree of class imbalance in the datasets range from 0.35 to 0.84, where the *Saccharomyces cerevisiae* datasets have the highest degree of class imbalance and the *Caenorhabditis elegans* datasets have the lowest degree of class imbalance.

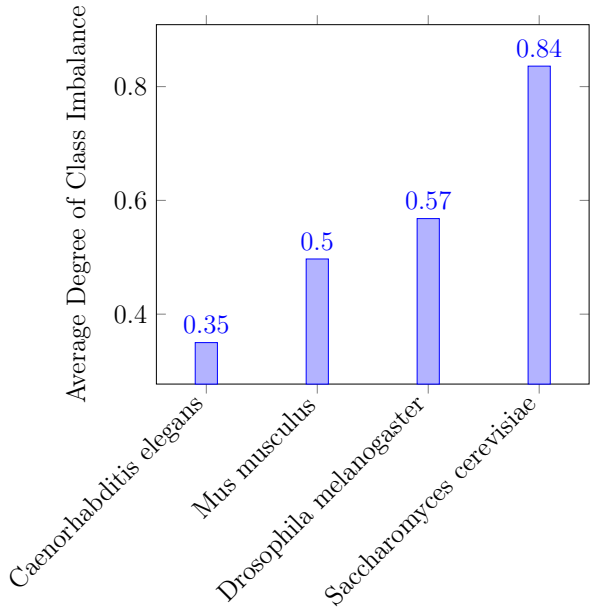
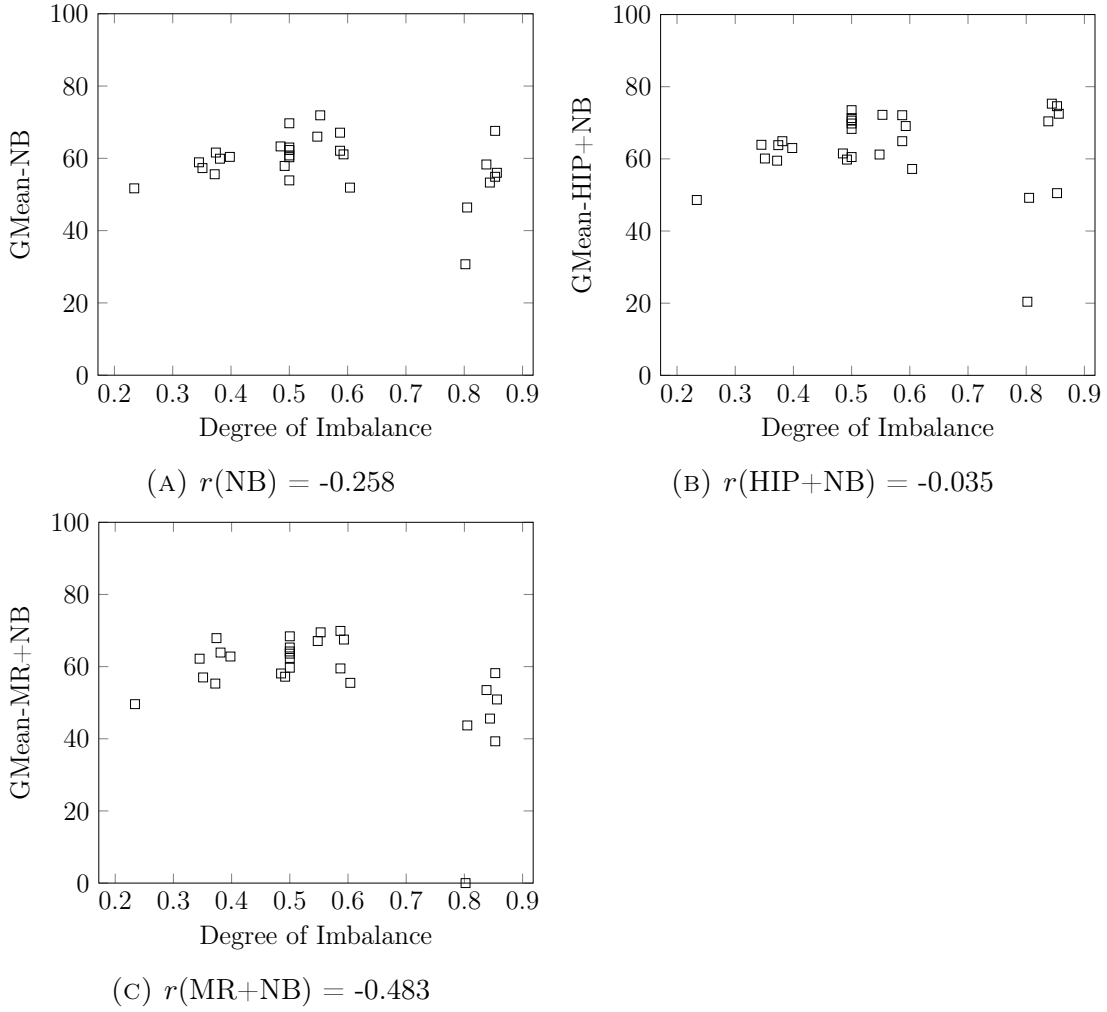


FIGURE 4.5 Average Degree of Class Imbalance for Each of the 4 Model Organisms Datasets – Averaged over the 7 Dataset Types

We calculated the linear correlation coefficient r between the degree of class imbalance and GMean values, as shown in Figure 4.6. HIP also shows the best performance compared with MR and Naïve Bayes without feature selection, because the value of the correlation coefficient for HIP is only -0.035, very close to 0, which means that HIP is only slightly affected by the degree of class imbalance in the datasets. In addition, it is worth noticing that MR shows the worst performance on the datasets where the degree of class imbalance is high, since MR has a relatively large negative correlation-coefficient value. The reason for this seems to be related with the nature of the predictive performance measure, i.e. the GMean measure, as follows.

In general, it can be observed in Tables 4.10 – 4.13 and 4.14 – 4.17 that both HIP and MR tend to obtain considerably higher Spe. than Sen. in the



(A) r(NB) = -0.258

(B) r(HIP+NB) = -0.035

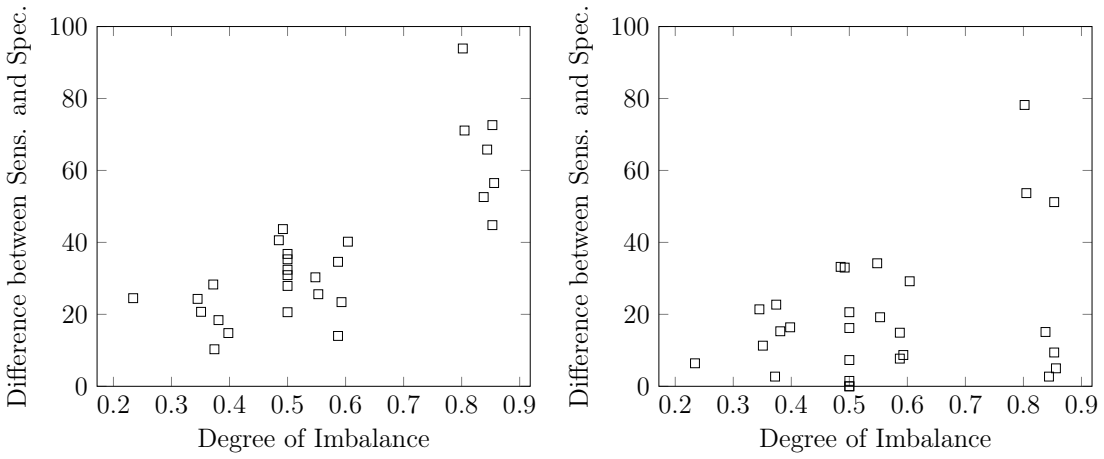
(C) r(MR+NB) = -0.483

FIGURE 4.6 Values of the Correlation Coefficient between the Degree of Class Imbalance in the Datasets and the GMean Value Obtained by HIP, MR and No Feature Selection

Caenorhabditis elegans and *Saccharomyces cerevisiae* datasets, where Spe. is a measure of predictive accuracy for instances belonging to the majority class ("Anti-Longevity"); and *vice versa*, both methods tend to obtain considerably higher Sen. than Spe. in the *Drosophila melanogaster* and *Mus musculus* datasets, where Sen. is a measure of predictive accuracy for instances belonging to the majority class ("Pro-Longevity"). It can also be observed in Tables 4.10 – 4.13 and 4.14 – 4.17 that, in general, the difference between Sen. and Spe. (i.e. **Diff**, calculated by Equation 4.6) is considerably larger for MR than for HIP. Hence, MR favours more strongly the prediction of the majority class, by comparison with HIP.

$$\mathbf{Diff} = \mathbf{Max}(Sen, Spe) - \mathbf{Min}(Sen, Spe) \tag{4.6}$$

We further calculated the linear correlation coefficient between **Diff** and the degree of class imbalance as shown in Figure 4.7. It is clear that MR has a much higher positive r value ($r = 0.790$) than HIP ($r = 0.332$), which indicates that a higher degree of class imbalance will lead to a **Diff** substantially larger for MR than for HIP. Recall that $GMean = \sqrt{Sen. \times Spe.}$, which means that GMean favours the balance between Sen. and Spe. Therefore, it can be concluded that HIP, which tends to select features that would lead to a considerably smaller difference between Sen. and Spe. than the MR method, shows stronger robustness against a large degree of class imbalance, contributing to HIP achieving in general higher GMean than MR.



(A) r(MR+NB) = 0.790 (B) r(HIP+NB) = 0.332

FIGURE 4.7 Value of the Correlation Coefficient between the Degree of Class Imbalance in the Datasets and the Difference between Sen. and Spe. for MR and HIP with Naïve Bayes

4.7.4.3 Comparing HIP and MR When Working with NB

We further compared the experiment results between HIP and MR methods. Table 4.19 shows the GMean values obtained by HIP/MR methods respectively working with NB for different datasets.

As shown by the boldface figures, which denote the higher value of GMean between the two methods, HIP outperforms MR 24 out of 28 times, while MR outperforms HIP 4 out of 28 times. We also conducted the statistical significance test (i.e. two-tailed Wilcoxon signed-rank test at 0.05 of significance level) on the

GMean values, and the result reveals that HIP significantly outperforms MR when working with NB.

TABLE 4.19 Predictive Accuracy for Naïve Bayes with the Hierarchical HIP and MR Methods

Organism	***Caenorhabditis elegans Datasets***						
GO Types	BP	MF	CC	BP+MF	BP+CC	MF+CC	BP+MF+CC
HIP + NB	**63.9**	48.6	**59.5**	63.8	**64.9**	**60.1**	**63.0**
MR + NB	62.2	**49.6**	55.3	**67.9**	63.9	57.0	62.8
Organism	***Drosophila melanogaster Datasets***						
GO Types	BP	MF	CC	BP+MF	BP+CC	MF+CC	BP+MF+CC
HIP + NB	**57.2**	**60.5**	61.2	**64.9**	**69.1**	**72.2**	**72.1**
MR + NB	55.5	59.7	**67.1**	59.5	67.5	69.5	69.9
Organism	***Mus musculus Datasets***						
GO Types	BP	MF	CC	BP+MF	BP+CC	MF+CC	BP+MF+CC
HIP + NB	**71.3**	**59.8**	**61.5**	**70.6**	**69.8**	68.3	**73.5**
MR + NB	63.6	57.2	58.1	64.2	62.4	**68.4**	65.3
Organism	***Saccharomyces cerevisiae Datasets***						
GO Types	BP	MF	CC	BP+MF	BP+CC	MF+CC	BP+MF+CC
HIP + NB	**70.4**	**20.4**	**49.2**	**75.3**	**74.6**	**50.5**	**72.5**
MR + NB	53.5	0.0	43.7	45.6	58.2	39.3	50.9

4.7.4.4 Scalability of Computational Running Time for Different Feature Selection Methods

In this section we report results about the computational time of the feature selection methods used in our experiments, to evaluate their scalability. It should be noted that many classification applications (including the ageing-related application in this thesis) are off-line, batch tasks, rather than online tasks; and the time spent collecting and preparing the data for classification typically is substantially greater than the time taken to run the classification algorithm. However, as one of criteria for evaluating classification methods, in any case, it is still interesting to investigate the scalability of the computational time taken by the feature selection methods used in our experiments, and to observe whether the time taken by running a feature selection method and then running NB with the selected features is smaller than the time taken to run NB with all original features (without using a feature selection method).

Hence, we measured the scalability of computational running time (using seconds as the unit of measure) for all feature selection methods and the combinations of those feature selection methods with the Naïve Bayes classifier. In order to measure this kind of scalability, we focus on measuring the computational time taken by each method in two datasets, namely the dataset with the smallest dimensionality and the dataset with the largest dimensionality. This shows the widest range of computational time for the datasets used in our experiments.

We measured the dimensionality **D** of a dataset by using Equation 4.7, which

$$\mathbf{D}(Dataset) = \mathbf{No.}(Features) \times \mathbf{No.}(Instance) \tag{4.7}$$

computes $\mathbf{D}(Dataset)$ in terms of the product of the number of features times the number of instances in the dataset. Equation 4.7 was applied to each of the 28 datasets used in our previous experiments (referring to 4 model organisms times 7 feature set types). After computing the value of the datasets' dimensionalities, we observed that the dataset for *Caenorhabditis elegans* consisting of biological process, molecular function, and cellular component (BP+MF+CC) GO terms as the features is the dataset having the largest dimensionality, while the dataset for *Drosophila melanogaster* consisting of only cellular component (CC) GO terms as

the features is the dataset having the smallest dimensionality. More precisely, the former dataset has dimensionality 681,252, whilst the latter dataset has dimensionality 6,750.

In addition, we estimated the computational running time of all algorithms working on those two datasets using the following approaches. Firstly, in terms of the dataset for *Caenorhabditis elegans* with BP+MF+CC GO terms, we only use 10 instances to run Naïve Bayes, and all lazy learning-based feature selection methods, which are high time-consuming. Then the average running time per instance (averaged over 10 instances) is obtained. This average running time per instance will be multiplied by 572, which is the total number of instances in that dataset. There is one exception for CFS/CFS+NB, which will run by using all instances in one cross validation fold (i.e. 58 instances). Then the average running time of CFS/CFS+NB per instance will be multiplied by 572 to estimate the total running time on the whole dataset.

Secondly, in terms of the dataset for *Drosophila melanogaster*, all methods will run by using all instances in one cross validation fold, i.e. 9 instances, in order to compute the average running time per instance. Then the estimated running time equals to the average running time per instance multiplied by 90, which is the total number of instances in the *Drosophila melanogaster* dataset. The use of these relatively small samples of instances, rather than using all instances, was chosen in order to avoid a very large computational time in the experiments to estimate scalability - these experiments were carried out separately, after the completion of the experiments that measured predictive accuracy.

Thirdly, the computer used for the experiments on estimating the computational time of all algorithms was an iMac equipped with one 2.9 GHz Intel Core i5 CPU, 2×4 GB 1600 MHz DDR3 memory, one Macintosh hard drive and OS X (version 10.8.2) operating system.

The results of these scalability experiments are included in Tables 4.20 – 4.21, where Table 4.20 shows the running time for different feature selection methods by themselves without including the time taken by the classification algorithm and Table 4.21 shows the running time for different feature selection methods working with Naïve Bayes classifier, i.e. the time taken to run both these types of methods together, as a whole. Overall, in Table 4.20, in the experiments with the *Caenorhabditis elegans* dataset, MR is the most time consuming method, taking

26,125 seconds (about 7.3 hours); while Ent_{HIP-k} is the least time consuming one, taking only 36.1 seconds. Comparing MR with HIP, the latter shows significantly better time efficiency, since the MR method conducts the feature selection process by comparing the relevance values of features for every individual path of the DAG, whereas the HIP method conducts the feature selection process merely by considering the hierarchical dependencies between features.

TABLE 4.20 Estimated Scalability of Computational Time (in Seconds) for Each Feature Selection Method

Datasets	**Algorithms**			
C. elegans ***(BP+MF+CC)*** $\mathbf{D}(Dataset) = 681252$	HIP	Ent_{HIP-k}	Rele_{HIP-k}	CFS
	1,158.2	36.1	40.6	16,637.3
	MR	Ent_{MR-k}	Rele_{MR-k}	
	26,125.0	36.7	40.4	
D. melanogaster ***(CC)*** $\mathbf{D}(Dataset) = 6750$	HIP	Ent_{HIP-k}	Rele_{HIP-k}	CFS
	214.2	2.5	3.7	2.2
	MR	Ent_{MR-k}	Rele_{MR-k}	
	224.6	2.6	3.7	

CFS is another very time-consuming method. On one hand, CFS is an eager method, which avoids the time consuming approach of selecting a different features set for each instance, like lazy methods. On the other hand, CFS adopts the *Backward–Greedy–Stepwise* searching approach to find the most appropriate subset of features. This search process is time consuming, especially in the dataset for *Caenorhabditis elegans* with BP+MF+CC GO terms as features, containing 1,191 candidate features.

Analogously to the results for the *Caenorhabditis elegans* dataset, in the *Drosophila melanogaster* dataset, consisting only of cellular component (CC) GO terms as the

features, MR is again the most time consuming method, and Ent_{HIP-k} is again the least time consuming one. One significant difference is that CFS works much more efficiently, i.e. it takes only 2.2 seconds. The reason is that the number of candidate features is only 75.

TABLE 4.21 Estimated Scalability of Computational Time (in Seconds) for Each Feature Selection Method Combined with Naïve Bayes

Datasets	**Algorithms**			
C. elegans	NB	HIP + NB	Ent_{HIP-k} + NB	Rele_{HIP-k} + NB
	1,887.6	1,272.6	211.2	212.0
(BP+MF+CC)	CFS + NB	MR + NB	Ent_{MR-k} + NB	Rele_{MR-k} + NB
$\mathbf{D}(Dataset) = 681252$	16,647.1	29,442.6	697.5	696.5
D. melanogaster	NB	HIP + NB	Ent_{HIP-k} + NB	Rele_{HIP-k} + NB
	0.5	215.0	2.8	3.9
(CC)	CFS + NB	MR + NB	Ent_{MR-k} + NB	Rele_{MR-k} + NB
$\mathbf{D}(Dataset) = 6750$	2.6	224.8	2.9	3.9

Table 4.21 shows the estimated computational running time for different feature selection methods working with the Naïve Bayes classifier. Analogously to the results shown in Table 4.20, in the experiments with the *Caenorhabditis elegans* dataset, MR+NB is the most time-consuming algorithm, whereas Ent_{HIP-k}+NB is the least time-consuming one. Note that the time taken by both CFS+NB and MR+NB is much greater than the time taken by NB without feature selection in the *Caenorhabditis elegans* dataset. However, for the other 5 feature selection methods, the time taken to run the feature selection method and then run NB with

the selected features is smaller than the time taken to run Naïve Bayes without feature selection.

In the experiments with the *Drosophila melanogaster* dataset, MR+NB is still the most time-consuming algorithm, whereas Naïve Bayes without any feature selection is the least time-consuming one. The large difference in the running time of Naïve Bayes without feature selection between the two different datasets is due to the large difference in the number of features between these datasets. More precisely, Naïve Bayes without feature selection is much slower on *Caenorhabditis elegans* dataset, since the number of Conditional Probability Tables (CPTs) is 1,191 (each CPT includes learnt parameters about one feature), whereas in the experiments with the *Drosophila melanogaster* dataset, the number of CPTs is only 75.

Overall, the estimated computational running time of different feature selection methods combined with the Naïve Bayes classifier reflects the scalability of time spent on experiments for different datasets, i.e. ranging from 0.5 second for the fastest method on the smallest dataset to 8.2 hours for the slowest method on the largest dataset.

Chapter 5

Lazy Hierarchical Feature Selection Methods with Tree Augmented Naïve Bayes

5.1 Introduction

In this chapter, we propose a lazy hierarchical feature selection method based on the Tree Augmented Naïve Bayes (TAN) classifier, called Hierarchy-based Redundancy Eliminated-Tree Augmented Naïve Bayes (HRE–TAN). Unlike the HIP and MR methods proposed in Chapter 4, which are filter feature selection methods, the proposed HRE–TAN is a type of embedded feature selection method. In this chapter we also compare the predictive accuracy of HRE–TAN with the accuracy of TAN using other feature selection methods in a pre-processing phase.

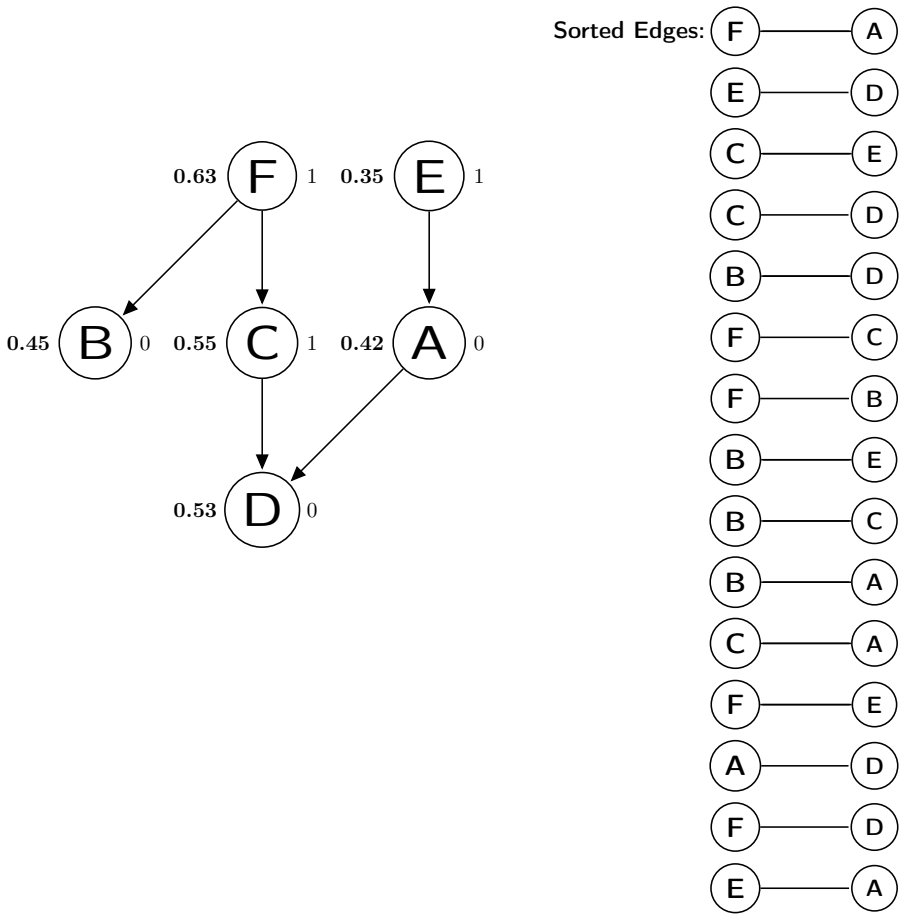


FIGURE 5.1 Example of a Small DAG of Features

5.2 Lazy Hierarchy-Based Redundancy Eliminated Tree Augmented Naïve Bayes (HRE–TAN)

This is a new method based on the lazy learning approach, and it performs embedded hierarchical feature selection, rather than conducting hierarchical feature selection in a pre-processing step. As mentioned in Chapter 2, a conventional TAN method builds a Maximum Weight Spanning Tree (MST) to detect dependencies among features, but it assumes that the feature are "flat", not hierarchical. In contrast, the proposed algorithm aims to eliminate the hierarchical redundancy between features when it builds the MST.

As discussed in Chapter 2, two features are hierarchically redundant if one of them is an ancestor or descendant of the other and they have the same feature value ("1" or "0"). To avoid the selection of hierarchically redundant features, HRE–TAN checks the status of each edge before adding it into the Undirected Acyclic Graph (UDAG). The status of an edge will be set to "Unavailable", if either of the vertices connected by the edge is hierarchically redundant, with respect to the vertices that have already been included in the UDAG, which will be transformed into a tree later by marking directions of edges. To describe how HRE–TAN works, we use the pseudocodes shown in Algorithms 5.1 and 5.2, as explained next.

In Algorithm 5.1, in the first part of the HRE–TAN algorithm (lines: 1–12), HRE–TAN firstly generates the Directed Acyclic Graph (DAG) for the current dataset; then it generates the set of ancestor and descendant features for each feature x_i. $\mathbf{Status}_{<E>}(x_i, x_j)$, which is initialised as "Available", denotes the selection status of the edge connecting vertices x_i and x_j. $\mathbf{CMI}_{<E>}(x_i, x_j)$ denotes the value of the conditional mutual information for the edge $\mathbf{E}(x_i, x_j)$. All edges are sorted in descending order of their conditional mutual information value (a greater value of conditional mutual information means a higher priority of adding the corresponding edge into the UDAG).

In the second part of the HRE–TAN algorithm (i.e. lines 13–21), the tree $\mathbb{T}$ will be built for each individual instance (i.e. adopting the lazy learning approach) by finding the Hierarchy-based Redundancy Eliminated-Maximum Weight Spanning Tree (HRE-MST). Then the training dataset and the current testing instance will be re-created with the features included in the tree, so that only those features will be used for classifying the re-created testing instance.

Algorithm 5.2 shows the pseudocode for building the HRE-MST. $\mathbf{NR}(x_i, x_j, \mathbf{Inst}_{<w>}, \mathbf{DAG})$ is a Boolean function that returns "True" if features x_i and x_j are non-hierarchically-redundant in the current testing instance $\mathbf{Inst}_{<w>}$, given the feature DAG. $\mathbf{NoCycle}(\mathbf{E}(x_i, x_j), \mathbf{UDAG})$ is a Boolean function that returns "True" if there is no cycle in the **UDAG** after adding edge $\mathbf{E}(x_i, x_j)$.

If the edge satisfies all the conditions in line 3 of Algorithm 5.2, it will be added into the **UDAG** (line 4). Once the algorithm has added the edge $\mathbf{E}(x_i, x_j)$ to the **UDAG**, for each of the two nodes connected by that edge, denoted as x_g (line 5), the algorithm will consider each of the nodes which are either an ancestor or a descendant of x_g in the feature **DAG**, denoting each such ancestor/descendant

as x_h (line 6). If feature x_g and its ancestor/descendant feature x_h have the same value in the current testing instance $\mathbf{Inst}_{<w>}$ (line 7), indicating a hierarchical redundancy in that pair of features, then the *for each* loop in lines 8–10 will set to "Unavailable" the status of all edges where one of the nodes is x_h – line 8, where the symbol "$*$" is a wildcard matching any node. In other words, among the set of hierarchically-redundant nodes (features) with the same value, HRE–TAN selects the node included in the edge having higher conditional mutual information, since Algorithm (5.2) processes edges in descending order of conditional mutual information.

To further explain how Algorithms 5.1 and 5.2 work, we use the example DAG shown in Figure 5.1, where the left part is a feature hierarchy consisting of three paths from a root to a leaf node of the **DAG**, i.e. from node F to node B; from node F to node D; and from node E to node D. The right part of Figure 5.1 shows the edges (for all pair of nodes) in descending order of **CMI**. HRE–TAN firstly adds edge $\mathbf{E}(F, A)$ into the UDAG, since its selection status is "*Available*"; nodes F and A are not hierarchically-redundant; and there is no cycle in the **UDAG** after adding edge $\mathbf{E}(F, A)$. Then, after adding $\mathbf{E}(F, A)$, Algorithm 5.2 will delete all edges that consist of the hierarchically redundant nodes with respect to either node F or node A, in order to eliminate the redundancy. Node C is redundant with respect to node F, because both of them have value "1" and are located in the same path in Figure 5.1. So, all edges containing node C (i.e. $\mathbf{E}(C, E)$, $\mathbf{E}(C, D)$, $\mathbf{E}(F, C)$, $\mathbf{E}(B, C)$, and $\mathbf{E}(C, A)$) will be unavailable to be added into the **UDAG**. Also, node D is redundant with respect to node A, because both of them have value "0" and are located in the same path. Thus, all edges consisting of node D (i.e. $\mathbf{E}(E, D)$, $\mathbf{E}(C, D)$, $\mathbf{E}(B, D)$, $\mathbf{E}(A, D)$ and $\mathbf{E}(F, D)$) will be unavailable to be added into the **UDAG**. Note that this hierarchical redundancy elimination process will dramatically reduce the size of the search space of candidate TAN structures.

After edges with node C or D had their selection status set to "*Unavailable*", edge $\mathbf{E}(F, B)$ – the next one available in the sorted list – will be added into the **UDAG**, since nodes F and B are not redundant (although both of them are in the same path in Figure 5.1, their values are different), and there is no cycle in the **UDAG** after adding that edge. Node B is not redundant with respect to any other node, so no edge has its status set to "Unavailable" in this step. Then, $\mathbf{E}(B, E)$ will

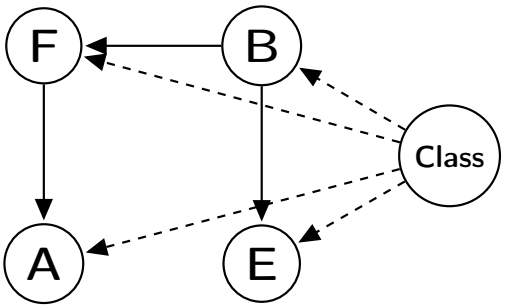


FIGURE 5.2 Example of Built HRE–MST Corresponding to Example in Figure 5.1

be added into the **UDAG** as the next available edge in the sorted edge list, since this edge also satisfies all conditions in line 3 of Algorithm 5.2. Then, $\mathbf{E}(B, A)$, $\mathbf{E}(F, E)$ and $\mathbf{E}(E, A)$ will be processed in turn. However, none of them will be added into the tree, since there would exist a cycle if each of them was added into the tree. Finally, HRE–TAN randomly selects a node as the root, which is used to mark directions of all edges in order to build the HRE–MST. Figure 5.2 shows the tree built from the example DAG shown in Figure 5.1, by selecting node B as the root. After finding the HRE-MST (i.e. tree $\mathbb{T}$), the training dataset and current testing instance will be re-created, and the testing instance will be classified using the built tree (line 17 in Algorithm 5.1). Then the selection status of all edges will be re-assigned as "Available" in line 19 of Algorithm 5.1, as a preparation for processing the next testing instance.

Algorithm 5.1 Lazy Hierarchical Redundancy Eliminated Tree Augmented Naïve Bayes (HRE–TAN)

1: Initialise **DAG** with all features in Dataset;
2: Initialise **TrainSet**;
3: Initialise **TestSet**;
4: **for** each feature $x_i \in \mathbb{X}$ **do**
5: Initialise $\mathbb{A}(x_i)$ in **DAG**;
6: Initialise $\mathbb{D}(x_i)$ in **DAG**;
7: **end for**
8: **for** each $\mathbf{E}(x_i, x_j) \in \mathbb{E}$ **do**
9: Calculate $\mathbf{CMI}_{<E>}(x_i, x_j)$ using **TrainSet**;
10: Initialise $\mathbf{Status}_{<E>}(x_i, x_j) \leftarrow$ "*Available*";
11: **end for**
12: Sort all $\mathbf{E}(x_i, x_j) \in \mathbb{E}$ by descending order of **CMI**;
13: **for** each instance $\mathbf{Inst}_{<w>} \in$ **TestSet do**
14: $\mathbb{T}$ = HRE–MST(**DAG**, $\mathbf{Inst}_{<w>}$, $\mathbb{A}(\mathbb{X})$, $\mathbb{D}(\mathbb{X})$, $\mathbb{E}$);
15: Re-create $\mathbf{TrainSet\_T}$ with feature set $\mathbb{X}' \in \mathbb{T}$;
16: Re-create $\mathbf{Inst\_T}_{<w>}$ with feature set $\mathbb{X}' \in \mathbb{T}$;
17: Tree Augmented Naïve Bayes($\mathbb{T}$, $\mathbf{TrainSet\_T}$, $\mathbf{Inst\_T}_{<w>}$);
18: **for** each $\mathbf{E}(x_i, x_j) \in \mathbb{E}$ **do**
19: Re-assign $\mathbf{Status}_{<E>}(x_i, x_j) \leftarrow$ "*Available*";
20: **end for**
21: **end for**

Algorithm 5.2 Hierarchical Redundancy Eliminated Maximum Weight Spanning Tree (HRE–MST)
(*assuming all edges are sorted in descending order of Conditional Mutual Information*)

Initialise an Empty **UDAG**;
for each $\mathbf{E}(x_i, x_j) \in \mathbb{E}$ **do**
  if $\{\mathbf{Status}_{<E>}(x_i, x_j) =$ "*Available*"$\} \wedge$
    $\{\mathbf{NR}(x_i, x_j, \mathbf{Inst}_{<w>}, \mathbf{DAG})\} \wedge$
    $\{\mathbf{NoCycle}(\mathbf{E}(x_i, x_j), \mathbf{UDAG})\}$ **then**
    add $\mathbf{E}(x_i, x_j)$ into **UDAG**;
    for each x_g in $\{x_i, x_j\}$ **do**
      for each x_h in $\{\mathbb{A}(x_g) \cup \mathbb{D}(x_g)\}$ **do**
        if $\mathbf{Value}(x_g, \mathbf{Inst}_{<w>}) = \mathbf{Value}(x_h, \mathbf{Inst}_{<w>})$ **then**
          for each $\mathbf{E}(x_h, *)$ **do**
            $\mathbf{Status}_{<E>}(x_h, *) \leftarrow$ "*Unavailable*";
          end for
        end if
      end for
    end for
  end if
end for
Choose **Root** by Randomly selecting vertex x in **UDAG**;
Build the tree ($\mathbb{T}$) by marking direction of all edges from the **Root** outwards to other vertices;
Return $\mathbb{T}$;

5.3 Experiments

5.3.1 Datasets Used in the Experiments

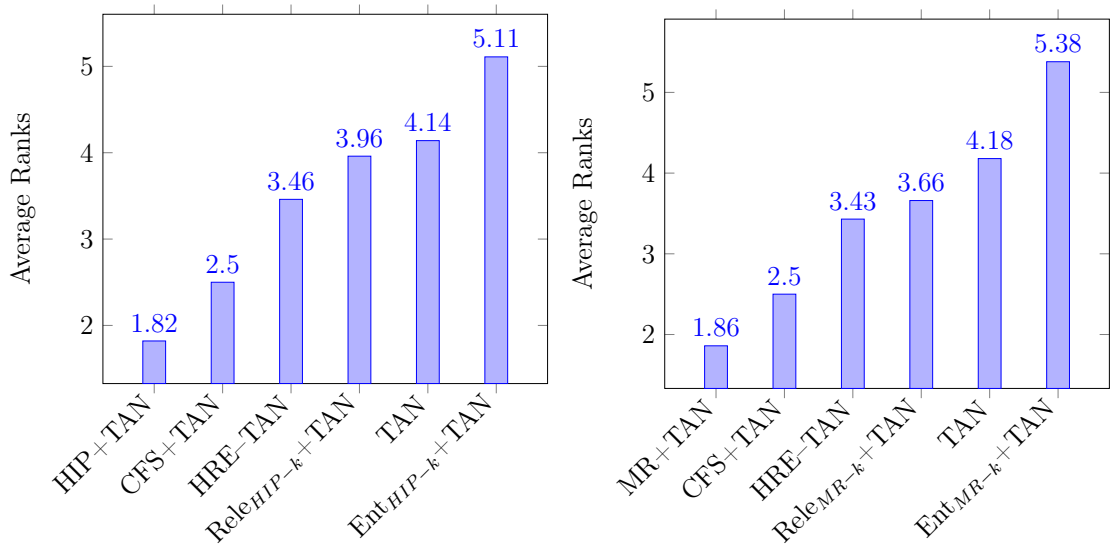
We evaluated the performance of feature selection methods using the same datasets mentioned in Section 4.7.2, i.e. for each model organism, we created 7 datasets, with all possible subsets of the three GO term types, i.e. one dataset for each type of GO term (BP, MF, CC), one dataset for each pair of GO term types (BP and MF, BP and CC, MF and CC), and one dataset with all 3 GO term types (BP, MF and CC).

5.3.2 Feature Selection Methods Evaluated in the Experiments

We compared HRE–TAN with the HIP and MR methods (the proposed hierarchical feature selection methods discussed on Chapter 4) when working with TAN. Briefly, either the MR or HIP method is used to perform lazy hierarchical feature selection in a pre-processing phase, before building a TAN structure. In essence, MR and HIP will substantially reduce the search space of candidate edges for building the Maximum Weight Spanning Tree (MST) used by TAN. This motivation is especially important for lazy learning algorithms, since they are computationally expensive. Apart from those hierarchical feature selection methods, we also experiment with the flat methods discussed in Chapter 4, which are Hybrid-lazy/eager-relevance-based feature selection (Rele_{HIP-k} and Rele_{MR-k}), Hybrid-lazy/eager-entropy-based feature selection (Ent_{HIP-k} and Ent_{MR-k}) and Correlation-based feature selection (CFS).

5.3.3 Experimental Results

Figure 5.3 summarises the average ranks (in terms of GMean values) of different feature selection methods working with TAN. Overall, either HIP or MR obtains the best rank compared with other feature selection methods. The results are shown in more detail, for each dataset, in Tables 5.1 through 5.8, as follows.



(A) Average Rank of Each Method for the Results Shown in Tables 5.1 – 5.4

(B) Average Rank of Each Method for the Results Shown in Tables 5.5 – 5.8

FIGURE 5.3 Summary of Ranks (*a lower value means a better predictive performance*) Based on GMean Values for Different Feature Selection Methods Working with TAN

Tables 5.1 – 5.4 compare the predictive accuracy of the hierarchical feature selection method HIP with the accuracies of Tree Augmented Naïve Bayes (TAN) using 3 "flat" feature selection methods (two Hybrid-lazy/eager methods, namely Ent$_{HIP-k}$, Rele$_{HIP-k}$, and one eager method, CFS), one lazy hierarchical feature selection method (Hierarchy-based Redundancy Eliminated TAN (HRE–TAN)), and TAN without using any feature selection method. In these tables, recall that GM stands for the geometric mean between sensitivity (Sen.) and specificity (Spe.), i.e. $GM = \sqrt{Sen. \times Spe.}$.

HIP+TAN ranks in the first position (average rank: 1.82) according to its GMean value and ranks first in 17 out of 28 datasets. The second best ranked method is CFS+TAN (average rank: 2.50), which successively ranks better than HRE–TAN (average rank: 3.46), Hybrid-lazy/eager-relevance-based (HIP-k)+TAN (average rank: 3.96), TAN without any feature selection (average rank: 4.14), and Hybrid-lazy/eager-entropy-based (HIP-k)+TAN (average rank: 5.11).

More precisely, in terms of results for each type of model organism, the main findings are as follows. In Table 5.1, for the datasets about *Caenorhabditis elegans*, the "anti-longevity" class is the majority class, and overall the values of specificity (a measure of accuracy for that majority class) are greater than sensitivity.

HIP+TAN obtains the highest GMean value 3 out of 7 times, CFS+TAN obtains the highest GMean value 2 times, while Hybrid-lazy/eager-entropy-based (HIP-k)+TAN and TAN without any feature selection obtain each the highest GMean value one time. In Table 5.2, for the datasets about *Drosophila melanogaster*, the "pro-longevity" class is the majority class, so the values of sensitivity (a measure of accuracy for that majority class) are greater than the values of specificity obtained by most of algorithms, except HIP+TAN working on BP, BP+MF, BP+CC and BP+MF+CC datasets, which obtains the highest GMean value 5 out of 7 times, while CFS+TAN and HRE–TAN obtain the highest GMean value one time each. In Table 5.3, for the datasets about *Mus musculus*, "pro-longevity" is the majority class, so overall the values of sensitivity are greater than the value of specificity obtained by most of algorithms, also existing some exceptions on HIP+TAN working on BP, BP+MF, BP+CC, MF+CC and BP+MF+CC datasets. HIP+TAN and CFS+TAN obtain the highest GMean value 2 out of 7 times, while Hybrid-lazy/eager-relevance-based (HIP-k)+TAN, HRE–TAN and TAN without any feature selection obtain the highest GMean value one time each. In Table 5.4, for the datasets about *Saccharomyces cerevisiae*, the "anti-longevity" class is the majority class, so the values of specificity are greater than the values of sensitivity obtained by all algorithms (except HIP+TAN working on BP+MF+CC dataset). Among those algorithms, HIP+TAN obtains the highest GMean value all 7 times.

Tables 5.5 – 5.8, which are analogous to Tables 5.1 – 5.4, compare the predictive accuracy of the hierarchical feature selection method MR with the accuracies of TAN using 3 "flat" feature selection methods (two lazy methods, namely Ent_{MR-k}, Rele_{MR-k}, and one eager method, CFS), one hierarchical feature selection method (Lazy Hierarchical Redundancy Eliminated TAN (HRE–TAN)), and TAN without feature selection. The MR+TAN also ranks in the first position and obtains the best ranks in 13 out of 28 datasets, with the average rank of 1.86, which is successively better than CFS+TAN (average rank: 2.50), HRE–TAN (average rank: 3.43), Hybrid-lazy/eager-relevance-based (MR-k)+TAN (average rank: 3.66), TAN without feature selection method (average rank: 4.18), and Hybrid-lazy/eager-entropy-based (MR-k)+TAN (average rank: 5.38).

In details, for the results about *Caenorhabditis elegans* datasets in Table 5.5, similarly to the results in Table 5.1, the values of specificity are greater than

the values of sensitivity obtained by all algorithms. MR+TAN obtains the highest GMean values 4 out of 7 times, while Hybrid-lazy/eager-entropy-based (MR-k)+TAN, CFS+TAN and TAN without any feature selection obtain the highest GMean value one time each. For the results about *Drosophila melanogaster* datasets in Table 5.6, similarly to the results in Table 5.2, the values of sensitivity are greater than the values of specificity obtained by all algorithms, while MR+TAN and CFS+TAN obtain the highest GMean value 3 out of 7 times, and HRE–TAN obtains the highest GMean value one time. For the results about *Mus musculus* datasets in Table 5.7, similarly to the results in Table 5.3, the values of sensitivity are greater than the values of specificity obtained by all algorithms, while MR+TAN obtains the highest GMean value 4 out of 7 times, HRE–TAN obtains the highest GMean value 2 times, and TAN without feature selection obtains the best result only one time. For the results about *Saccharomyces cerevisiae* datasets in Table 5.8, similarly to the results in Table 5.4, the values of specificity are greater than the values of sensitivity obtained by all algorithms, while CFS+TAN obtains the highest GMean values 4 out of 7 times, MR+TAN obtains the highest value 2 times, Hybrid-lazy/eager-relevance-based (MR-k)+TAN obtains the highest value one time as a draw with HRE–TAN.

TABLE 5.1 Predictive Accuracy for Tree Augmented Naïve Bayes with the Hierarchical HIP Method and Baseline "Flat" Feature Selection Methods on *Caenorhabditis elegans* Datasets

Caenorhabditis elegans Datasets																				
Feature		**TAN without**			**Lazy HIP + TAN**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + TAN**			**Lazy HRE – TAN**		
Type		**Feature Selection**						**Ent_{HIP-k} + TAN**			**Rele_{HIP-k} + TAN**								
		Sens.	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*
	BP	34.0±3.2	79.6±2.3	52.0	52.2±2.3	67.7±3.5	59.4	34.9±3.4	84.3±1.7	54.2	32.1±2.3	83.1±2.3	51.6	45.9±3.7	79.3±2.2	**60.3**	41.1±2.4	76.8±2.1	56.2
	MF	37.2±5.8	61.4±5.0	47.8	43.0±5.6	50.6±4.5	46.6	38.0±5.1	66.5±5.2	**50.3**	15.7±3.6	82.9±3.3	36.1	24.8±4.8	74.7±4.0	43.0	23.1±4.8	75.3±5.4	41.7
	CC	39.8±3.0	78.2±2.2	**55.8**	44.9±2.7	62.2±4.7	52.8	28.6±5.0	80.8±3.0	48.1	29.6±4.0	76.9±3.6	47.7	34.7±4.3	76.9±3.2	51.7	24.5±3.6	80.8±3.0	44.5
	BP+MF	35.2±1.9	80.3±2.2	53.2	54.5±3.2	72.1±2.4	**62.7**	38.0±4.3	82.1±1.5	55.9	35.2±3.4	82.6±2.1	53.9	46.0±3.2	80.6±2.0	60.9	42.3±2.3	80.0±2.6	58.2
	BP+CC	42.7±3.1	81.7±2.7	59.1	59.2±3.9	69.2±2.9	**64.0**	42.3±3.3	82.3±2.3	59.0	35.2±2.4	83.7±1.9	54.3	45.1±2.8	80.8±2.0	60.4	44.6±3.0	74.4±3.6	57.6
	MF+CC	40.6±3.4	74.4±3.6	55.0	45.3±2.2	67.2±3.5	55.2	37.6±3.2	74.4±3.5	52.9	39.4±3.7	75.2±3.4	54.4	47.1±3.5	73.7±3.5	**58.9**	32.4±3.3	79.8±3.2	50.8
	BP+MF+CC	39.5+2.8	80.1+2.6	56.2	60.0+5.5	71.4+2.2	**65.5**	37.7+2.7	79.0+1.7	54.6	39.5+4.1	81.0+1.7	56.6	45.6+5.0	77.3+2.2	59.4	44.2+3.9	79.3+2.9	59.2

TABLE 5.2 Predictive Accuracy for Tree Augmented Naïve Bayes with the Hierarchical HIP Method and Baseline "Flat" Feature Selection Methods on *Drosophila melanogaster* Datasets

Drosophila melanogaster Datasets																				
Feature		**TAN without**			**Lazy HIP + TAN**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + TAN**			**Lazy HRE – TAN**		
Type		**Feature Selection**						**Ent_{HIP-k} + TAN**			**Rele_{HIP-k} + TAN**								
		Sens.	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*
BP		92.3±2.9	19.4±8.4	42.3	58.2±6.5	72.2±5.4	**64.8**	94.5±1.9	2.8±2.5	16.3	85.7±3.3	25.0±5.9	46.3	79.1±5.1	25.0±5.9	44.5	86.8±3.2	30.6±10.2	51.5
MF		91.2±3.3	20.6±5.0	43.3	73.5±5.5	32.4±7.1	48.8	91.2±3.3	26.5±6.0	49.2	91.2±2.5	35.3±7.2	56.7	85.3±4.3	32.4±7.1	52.6	86.8±3.4	41.2±8.8	**59.8**
CC		90.3±3.6	32.1±11.6	53.8	79.0±3.6	50.0±11.3	**62.8**	95.2±2.4	25.0±7.1	48.8	90.3±2.6	35.7±9.9	56.8	87.1±3.8	42.9±10.2	61.1	75.8±5.8	28.6±9.7	46.6
BP+MF		92.4±3.3	23.7±6.9	46.8	52.2±4.0	73.7±5.8	**62.0**	96.7±2.4	13.2±4.2	35.7	85.9±4.1	28.9±7.9	49.8	85.9±2.9	31.6±5.3	52.1	87.0±3.3	31.6±6.5	52.4
BP+CC		86.8±4.0	18.9±7.6	40.5	59.3±5.7	67.6±7.2	**63.3**	94.5±1.8	8.1±4.7	27.7	82.4±3.7	29.7±8.5	49.5	79.1±5.0	48.6±10.4	62.0	84.6±2.4	32.4±10.6	52.4
MF+CC		90.6±3.3	31.6±5.0	53.5	76.5±4.9	60.5±9.3	68.0	96.5±2.3	28.9±6.9	52.8	92.9±2.5	39.5±5.5	60.6	89.4±3.8	52.6±5.8	**68.6**	87.1±4.4	39.5±5.5	58.7
BP+MF+CC		92.4±2.4	18.4±5.3	41.2	60.9±7.6	78.9±6.9	**69.3**	97.8±1.5	13.2±6.7	35.9	91.3±2.2	42.1±8.4	62.0	85.9±1.8	47.4±8.7	63.8	82.6±3.4	47.4±8.7	62.6

TABLE 5.3 Predictive Accuracy for Tree Augmented Naïve Bayes with the Hierarchical HIP Method and Baseline "Flat" Feature Selection Methods on *Mus musculus* Datasets

Mus musculus Datasets																				
Feature	**TAN without**			**Lazy HIP + TAN**			**Lazy/Eager**			**Lazy/Eager**			**Eager CFS + TAN**			**Lazy HRE – TAN**		
Type	**Feature Selection**						**Ent**$_{HIP-k}$ **+ TAN**			**Rele**$_{HIP-k}$ **+ TAN**								
	Sens.	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*
BP	89.7±3.7	41.2±4.9	60.8	42.6±5.3	73.5±7.2	56.0	98.5±1.4	32.4±5.5	56.5	85.3±4.8	38.2±8.5	57.1	82.4±3.6	47.1±6.2	62.3	86.8±5.5	47.1±4.7	**63.9**
MF	89.2±4.0	33.3±9.4	54.5	69.2±7.7	66.7±7.6	**67.9**	86.2±2.6	36.4±12.9	56.0	89.2±3.8	30.3±9.5	52.0	86.2±4.0	30.3±9.6	51.1	83.1±3.3	42.4±9.3	59.4
CC	75.8±4.4	41.2±8.3	55.9	72.7±5.1	50.0±10.1	**60.3**	83.3±3.3	26.5±7.3	47.0	81.8±4.7	32.4±9.3	51.5	75.8±3.2	38.2±12.6	53.8	86.4±4.0	41.2±9.7	59.7
BP+MF	86.8±3.4	35.3±5.4	55.4	42.6±4.9	79.4±9.3	58.2	97.1±1.9	32.4±6.4	56.1	91.2±3.2	38.2±6.2	59.0	88.2±4.2	41.2±8.0	**60.3**	83.8±4.5	41.2±6.8	58.8
BP+CC	88.2±3.6	47.1±9.7	**64.5**	48.5±4.4	82.4±6.8	63.2	97.1±1.9	29.4±5.2	53.4	80.9±7.1	47.1±9.7	61.7	83.8±5.0	41.2±8.7	58.8	79.4±4.9	47.1±9.7	61.2
MF+CC	88.2±4.2	41.2±10.0	60.3	63.2±3.1	64.7±12.7	63.9	89.7±3.7	38.2±9.4	58.5	89.7±4.3	50.0±10.2	**67.0**	77.9±3.8	52.9±10.8	64.2	89.7±3.0	35.3±9.6	56.3
BP+MF+CC	91.2±3.2	41.2±8.6	61.3	45.6±8.0	82.4±5.2	61.3	94.1±2.3	35.3±8.4	57.6	89.7±3.0	41.2±7.9	60.8	77.9±4.9	55.9±7.0	**66.0**	85.3±3.7	44.1±8.9	61.3

TABLE 5.4 Predictive Accuracy for Tree Augmented Naïve Bayes with the Hierarchical HIP Method and Baseline "Flat" Feature Selection Methods on *Saccharomyces cerevisiae* Datasets

Saccharomyces cerevisiae Datasets																				
Feature Type	**Feature Selection**	**TAN without**			**Lazy HIP + TAN**			**Lazy/Eager Ent$_{HIP-k}$ + TAN**			**Lazy/Eager Rele$_{HIP-k}$ + TAN**			**Eager CFS + TAN**			**Lazy HRE – TAN**		
		Sens.	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*
BP		3.3±3.3	98.9±1.1	18.1	56.7±10.0	68.6±2.0	**62.4**	0.0±0.0	99.5±0.5	0.0	16.7±7.5	94.1±1.7	39.6	33.3±7.0	91.9±2.4	55.3	20.0±7.4	93.5±1.7	43.2
MF		0.0±0.0	97.7±1.2	0.0	26.9±6.2	78.6±2.7	**46.0**	0.0±0.0	98.5±1.0	0.0	0.0±0.0	96.2±1.3	0.0	5.0±5.0	94.7±1.2	21.8	0.0±0.0	96.9±1.7	0.0
CC		16.7±7.0	95.9±2.1	40.0	25.0±10.6	85.4±4.0	**46.2**	5.0±5.0	97.6±1.2	22.1	12.5±6.9	95.1±2.1	34.5	16.7±7.0	93.5±1.6	39.5	12.5±6.1	93.5±2.9	34.2
BP+MF		3.3±3.3	99.0±0.7	18.1	63.3±9.2	67.7±3.1	**65.5**	0.0±0.0	99.5±0.5	0.0	13.3±5.4	94.3±0.9	35.4	30.0±6.0	93.8±1.7	53.0	26.7±10.9	95.8±1.5	50.6
BP+CC		10.0±5.1	99.0±0.7	31.5	63.3±6.0	73.5±3.8	**68.2**	0.0±0.0	100.0±0.0	0.0	20.0±7.4	95.6±1.6	43.7	33.3±8.6	94.1±1.6	56.0	26.7±6.7	94.1±2.1	50.1
MF+CC		5.0±5.0	98.5±0.8	22.2	31.0±9.9	81.7±2.5	**50.3**	0.0±0.0	100.0±0.0	0.0	10.3±6.1	95.4±1.6	31.3	10.3±6.1	94.4±1.4	31.2	10.3±6.1	95.4±1.9	31.3
BP+MF+CC		0.0±0.0	99.0±0.6	0.0	70.0±9.2	69.7±3.0	**69.8**	0.0±0.0	99.5±0.5	0.0	16.7±7.5	93.8±1.3	39.6	33.3±9.9	91.8±2.1	55.3	23.3±7.1	96.2±1.4	47.3

TABLE 5.5 Predictive Accuracy for Tree Augmented Naïve Bayes with the Hierarchical MR Method and Baseline "Flat" Feature Selection Methods on *Caenorhabditis elegans* Datasets

Caenorhabditis elegans Datasets																				
Feature Type	**TAN without Feature Selection**			**Lazy MR + TAN**			**Lazy/Eager Ent$_{MR-k}$ + TAN**			**Lazy/Eager Rele$_{MR-k}$ + TAN**			**Eager CFS + TAN**			**Lazy HRE – TAN**		
	Sens.	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*
BP	34.0±3.2	79.6±2.3	52.0	55.0±2.4	73.0±1.8	**63.4**	30.1±3.8	83.7±2.8	50.2	36.8±3.1	80.6±2.4	54.5	45.9±3.7	79.3±2.2	60.3	41.1±2.4	76.8±2.1	56.2
MF	37.2±5.8	61.4±5.0	47.8	33.1±3.5	65.2±4.0	46.5	40.5±5.2	64.6±5.3	**51.1**	24.8±3.5	75.3±5.0	43.2	24.8±4.8	74.7±4.0	43.0	23.1±4.8	75.3±5.4	41.7
CC	39.8±3.0	78.2±2.2	**55.8**	37.8±3.4	74.4±2.7	53.0	30.6±3.5	76.9±3.6	48.5	33.7±5.4	75.0±2.8	50.3	34.7±4.3	76.9±3.2	51.7	24.5±3.6	80.8±3.0	44.5
BP+MF	35.2±1.9	80.3±2.2	53.2	61.0±4.3	71.8±2.3	**66.2**	37.1±4.1	83.8±1.7	55.8	43.7±3.6	81.2±2.7	59.6	46.0±3.2	80.6±2.0	60.9	42.3±2.3	80.0±2.6	58.2
BP+CC	42.7±3.1	81.7±2.7	59.1	56.3±3.0	77.3±2.2	**66.0**	34.7±4.5	82.6±2.5	53.5	44.1±2.1	82.6±1.3	60.4	45.1±2.8	80.8±2.0	60.4	44.6±3.0	74.4±3.6	57.6
MF+CC	40.6±3.4	74.4±3.6	55.0	45.9±3.8	70.6±3.0	56.9	35.9±3.2	73.7±2.9	51.4	40.0±3.3	74.0±3.4	54.4	47.1±3.5	73.7±3.5	**58.9**	32.4±3.3	79.8±3.2	50.8
BP+MF+CC	39.5+2.8	80.1+2.6	56.2	54.4+4.2	76.5+2.3	**64.5**	34.0+3.0	82.6+1.5	53.0	44.2+3.8	80.7+1.6	59.7	45.6+5.0	77.3+2.2	59.4	44.2+3.9	79.3+2.9	59.2

TABLE 5.6 Predictive Accuracy for Tree Augmented Naïve Bayes with the Hierarchical MR Method and Baseline "Flat" Feature Selection Methods on *Drosophila melanogaster* Datasets

Drosophila melanogaster Datasets																				
Feature Type	**Feature Selection**	**TAN without**			**Lazy MR + TAN**			**Lazy/Eager Ent$_{MR-k}$ + TAN**			**Lazy/Eager Rele$_{MR-k}$ + TAN**			**Eager CFS + TAN**			**Lazy HRE – TAN**		
		Sens.	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*
BP		92.3±2.9	19.4±8.4	42.3	76.9±3.6	50.0±9.6	**62.0**	95.6±2.5	2.8±2.5	16.4	87.9±3.8	27.8±7.5	49.4	79.1±5.1	25.0±5.9	44.5	86.8±3.2	30.6±10.2	51.5
MF		91.2±3.3	20.6±5.0	43.3	83.8±4.5	41.2±7.4	58.8	92.6±3.4	32.4±6.3	54.8	89.7±2.4	35.3±6.1	56.3	85.3±4.3	32.4±7.1	52.6	86.8±3.4	41.2±8.8	**59.8**
CC		90.3±3.6	32.1±11.6	53.8	75.8±6.6	42.9±8.3	57.0	95.2±2.4	25.0±7.1	48.8	88.7±4.3	35.7±9.9	56.3	87.1±3.8	42.9±10.2	**61.1**	75.8±5.8	28.6±9.7	46.6
BP+MF		92.4±3.3	23.7±6.9	46.8	80.4±2.8	47.4±9.5	**61.7**	96.7±2.4	13.2±4.2	35.7	87.0±3.6	23.7±6.9	45.4	85.9±2.9	31.6±5.3	52.1	87.0±3.3	31.6+6.5	52.4
BP+CC		86.8±4.0	18.9±7.6	40.5	82.4±3.8	40.5±8.0	57.8	94.5±2.3	10.8±5.2	31.9	83.5±4.3	27.0±9.0	47.5	79.1±5.0	48.6±10.4	**62.0**	84.6±2.4	32.4±10.6	52.4
MF+CC		90.6±3.3	31.6±5.0	53.5	72.9±6.4	52.6±6.9	61.9	96.5±2.4	23.7±6.9	47.8	92.9±2.5	42.1±3.8	62.5	89.4±3.8	52.6±5.8	**68.6**	87.1±4.4	39.5±5.5	58.7
BP+MF+CC		92.4±2.4	18.4±5.3	41.2	77.2±4.5	60.5±8.5	**68.3**	98.9±1.1	13.2±6.7	36.1	92.4±2.4	42.1±8.4	62.4	85.9±1.8	47.4±8.7	63.8	82.6±3.4	47.4±8.7	62.6

TABLE 5.7 Predictive Accuracy for Tree Augmented Naïve Bayes with the Hierarchical MR Method and Baseline "Flat" Feature Selection Methods on *Mus musculus* Datasets

***Mus musculus* Datasets**																				
Feature Type	**Feature Selection**	**TAN without**			**Lazy MR + TAN**			**Lazy/Eager Ent$_{MR-k}$ + TAN**			**Lazy/Eager Rele$_{MR-k}$ + TAN**			**Eager CFS + TAN**			**Lazy HRE – TAN**		
		Sens.	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*
BP		89.7±3.7	41.2±4.9	60.8	73.5±7.1	50.0±10.0	60.6	97.1±1.9	26.5±3.4	50.7	83.8±4.5	41.2±7.4	58.8	82.4±3.6	47.1±6.2	62.3	86.8±5.5	47.1±4.7	**63.9**
MF		89.2±4.0	33.3±9.4	54.5	83.1±6.6	54.5±9.1	**67.3**	89.2±3.2	33.3±12.5	54.5	87.7±3.6	39.4±11.2	58.8	86.2±4.0	30.3±9.6	51.1	83.1±3.3	42.4±9.3	59.4
CC		75.8±4.4	41.2±8.3	55.9	74.2±4.3	44.1±9.8	57.2	86.4±4.0	23.5±10.4	45.1	78.8±4.0	26.5±10.2	45.7	75.8±3.2	38.2±12.6	53.8	86.4±4.0	41.2±9.7	**59.7**
BP+MF		86.8±3.4	35.3±5.4	55.4	79.4±4.3	55.9±8.6	**66.6**	95.6±2.2	26.5±4.5	50.3	89.7±3.7	35.3±5.4	56.3	88.2±4.2	41.2±8.0	60.3	83.8±4.5	41.2±6.8	58.8
BP+CC		88.2±3.6	47.1±9.7	**64.5**	70.6±5.9	58.8±8.9	64.4	98.5±1.4	32.4±6.4	56.5	80.9±7.1	41.2±10.5	57.7	83.8±5.0	41.2±8.7	58.8	79.4±4.9	47.1±9.7	61.2
MF+CC		88.2±4.2	41.2±10.0	60.3	82.4±3.6	55.9±11.5	**67.9**	92.6±3.2	38.2±9.4	59.5	88.2±4.7	50.0±10.2	66.4	77.9±3.8	52.9±10.8	64.2	89.7±3.0	35.3±9.6	56.3
BP+MF+CC		91.2±3.2	41.2±8.6	61.3	75.0±5.7	58.8±7.9	**66.4**	94.1±2.3	29.4±6.4	52.6	89.7±3.0	41.2±9.9	60.8	77.9±4.9	55.9±7.0	66.0	85.3±3.7	44.1±8.9	61.3

TABLE 5.8 Predictive Accuracy for Tree Augmented Naïve Bayes with the Hierarchical MR Method and Baseline "Flat" Feature Selection Methods on *Saccharomyces cerevisiae* Datasets

Saccharomyces cerevisiae Datasets																				
Feature Type		**TAN without Feature Selection**			**Lazy MR + TAN**			**Lazy/Eager Ent$_{MR-k}$ + TAN**			**Lazy/Eager Rele$_{MR-k}$ + TAN**			**Eager CFS + TAN**			**Lazy HRE – TAN**		
		Sens.	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*	*Sens.*	*Spec.*	*GM*
BP		3.3±3.3	98.9±1.1	18.1	30.0±7.8	87.0±2.7	51.1	0.0±0.0	100.0±0.0	0.0	10.0±5.1	93.0±3.1	30.5	33.3±7.0	91.9±2.4	**55.3**	20.0±7.4	93.5±1.7	43.2
MF		0.0±0.0	97.7±1.2	0.0	0.0±0.0	87.8±2.9	0.0	0.0±0.0	98.5±1.0	0.0	0.0±0.0	96.2±1.3	0.0	5.0±5.0	94.7±1.2	**21.8**	0.0±0.0	96.9±1.7	0.0
CC		16.7±7.0	95.9±2.1	40.0	20.8±6.9	95.1±2.1	**44.5**	5.0±5.0	95.9±1.8	21.9	12.5±6.9	94.3±2.4	34.3	16.7±7.0	93.5±1.6	39.5	12.5±6.1	93.5±2.9	34.2
BP+MF		3.3±3.3	99.0±0.7	18.1	20.0±7.4	93.2±1.4	43.2	0.0±0.0	100.0±0.0	0.0	16.7±5.6	95.3±1.5	39.9	30.0±6.0	93.8±1.7	**53.0**	26.7±10.9	95.8±1.5	50.6
BP+CC		10.0±5.1	99.0±0.7	31.5	30.0±9.2	89.2±2.1	51.7	6.7±4.4	100.0±0.0	25.9	13.3±5.4	94.1±1.6	35.4	33.3±8.6	94.1±1.6	**56.0**	26.7±6.7	94.1±2.1	50.1
MF+CC		5.0±5.0	98.5±0.8	22.2	10.3±6.1	93.4±2.5	31.0	6.9±5.7	99.5±0.5	26.2	10.3±6.1	95.4±1.8	**31.3**	10.3±6.1	94.4±1.4	31.2	10.3±6.1	95.4±1.9	**31.3**
BP+MF+CC		0.0±0.0	99.0±0.6	0.0	36.7±9.2	89.4±2.1	**57.3**	3.3±3.3	100.0±0.0	18.2	10.0±5.1	96.6±1.3	31.1	33.3±9.9	91.8±2.1	55.3	23.3±7.1	96.2±1.4	47.3

5.4 Discussion

5.4.1 Statistical Analysis of GMean Value Differences between the Feature Selection Methods

The Friedman test and the Holm *post-hoc* method were adopted for conducting a statistical significance test on the differences of GMean values between the methods working on all 4 model organisms. The HIP and MR methods were chosen as the control methods, since each of them performed best among the methods compared in Tables 5.1 – 5.4 and 5.5 – 5.8, respectively.

The significance test results are listed in Table 5.9, where the left part of the table reports results for HIP and the right part reports results for MR. The control method (HIP or MR) is considered significantly better than another method if, in the row for that other method, the p-value is smaller than the adjusted α. The significant results are shown in boldface in Table 5.9. Both HIP and MR significantly improve the performance of conventional TAN without using feature selection, significantly outperform the Hybrid-lazy/eager-relevance-based feature selection method, the Hybrid-lazy/eager-entropy-based feature selection method, and the HRE–TAN method, but show a non-significant difference to CFS.

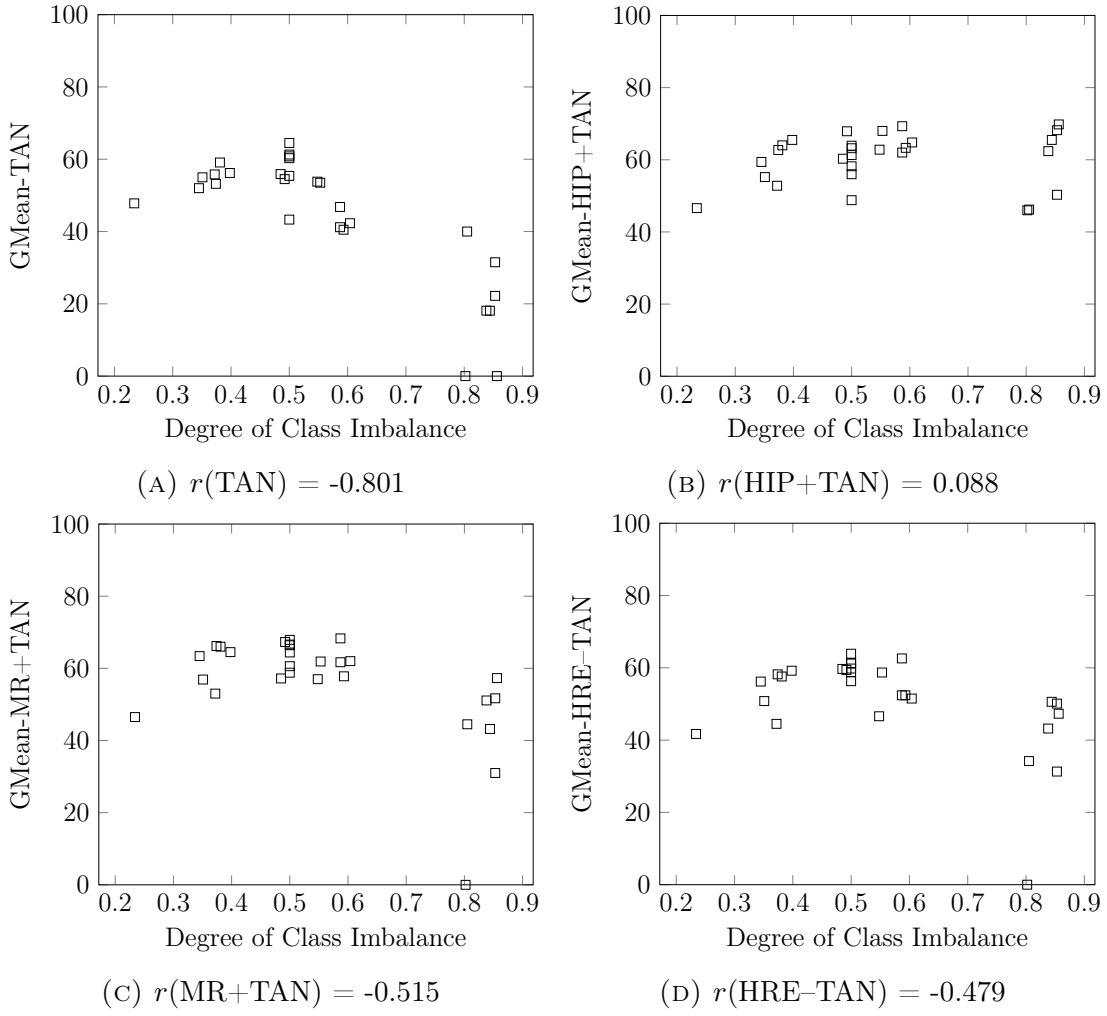
TABLE 5.9 Statistical Test Results of the Methods' GMean Values According to the Non-Parametric Friedman Test with the Holm *Post-Hoc* Test at the $\alpha = 0.05$ Significance Level

FS Method	**Ave. Rank**	**P-value**	**Adjusted α**	**FS Method**	**Ave. Rank**	**P-value**	**Adjusted α**
HIP (ctrl.)	1.82	–	–	MR (ctrl.)	1.86	–	–
CFS	2.50	1.74 E-01	0.0500	CFS	2.50	2.00 E-01	0.0500
HRE–TAN	3.46	**1.04 E-03**	0.0250	HRE–TAN	3.43	**1.69 E-03**	0.0250
Rele_{HIP-k}	3.96	**1.87 E-05**	0.0167	Rele_{MR-k}	3.66	**3.18 E-04**	0.0167
No FS	4.14	**3.48 E-06**	0.0125	No FS	4.18	**3.48 E-06**	0.0125
Ent_{HIP-k}	5.11	**4.70 E-11**	0.0100	Ent_{MR-k}	5.38	**1.92 E-12**	0.0100

5.4.2 Analysis of the Correlation between Degrees of Class Imbalance and GMean Values

We calculated the linear correlation coefficient r between the degrees of class imbalance in the datasets and the GMean values obtained by HIP and MR working with TAN, and HRE–TAN methods, as shown in Figure 5.4. The degree of imbalance was defined in Chapter 4, i.e. it is the complement of the ratio of the number of instances belonging to the minority class over the number of instances belonging to the majority class, as shown in Equation 4.5. For more details about the analysis of the correlation between the degree of class imbalance and the GMean value obtained by a feature selection method, see Section 4.7.4.2.

Similarly to the results obtained when working with Naïve Bayes (in Chapter 4), HIP still shows the strongest robustness against a large degree of class imbalance. The r value for HIP method is 0.088, which means that HIP is still little affected by the class imbalance issue. HRE–TAN obtains the second best r value, i.e. -0.479, which is better than the r value (i.e. -0.515) obtained by MR method. TAN without using feature selection obtains the worst r value (-0.801). Overall, all three hierarchical feature selection methods are able to enhance the robustness against the class imbalance issue for the TAN classifier, by comparison with no feature selection. However, among those three methods, HIP is still the best one in terms of robustness against class imbalance.



(A) r(TAN) = -0.801

(B) r(HIP+TAN) = 0.088

(C) r(MR+TAN) = -0.515

(D) r(HRE–TAN) = -0.479

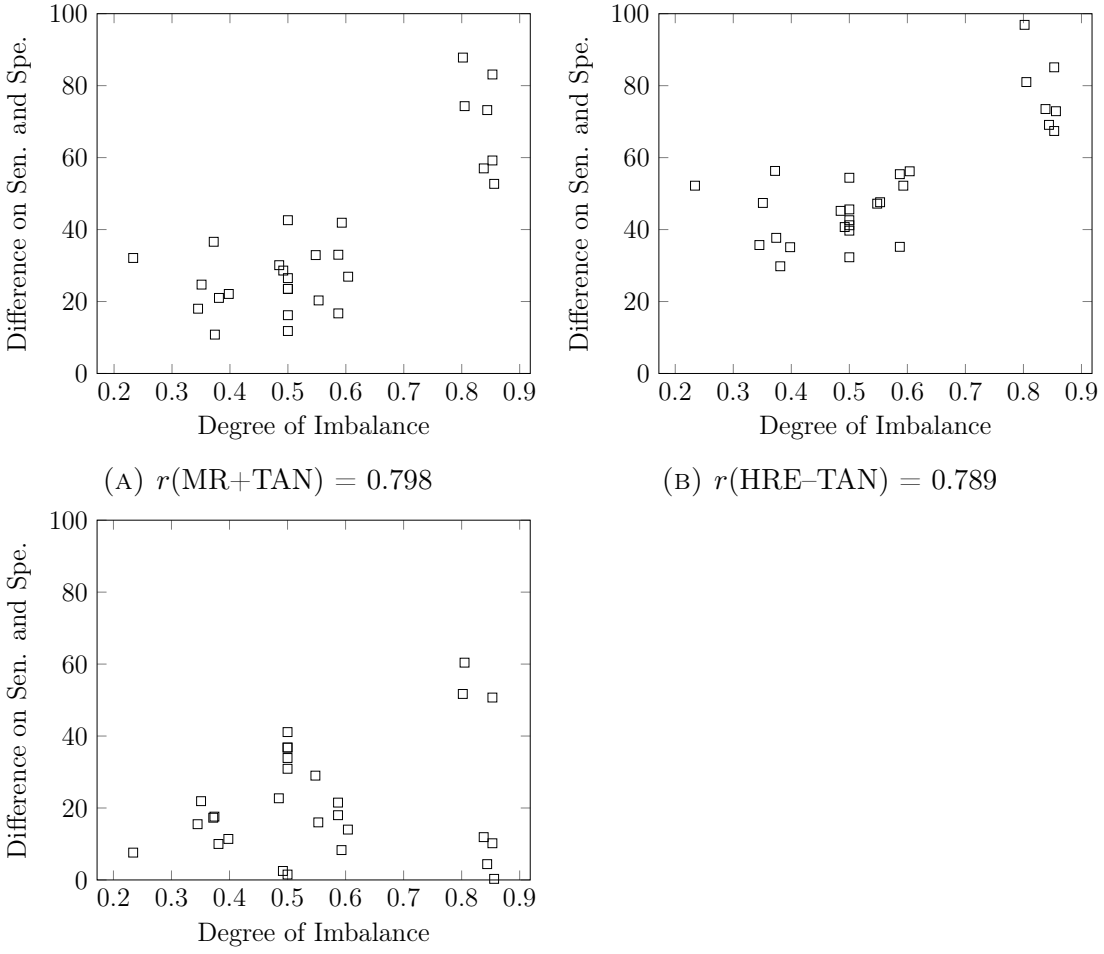
FIGURE 5.4 Values of the Correlation Coefficient (r) between the Degree of Class Imbalance and GMean Values for No Feature Selection with TAN, HIP+TAN, MR+TAN and HRE–TAN

5.4.3 Analysis of the Correlation between Degrees of Class Imbalance and Differences between Sen. and Spe.

In order to investigate the reason why HIP shows the strongest robustness against a large degree of class imbalance, we use here the same approach used in Section 4.7.4.2. Hence, we observed that the HIP+TAN's difference between Sen. and Spe. is smaller than MR+TAN's and HRE–TAN's difference. Then we calculated the linear correlation coefficient (r) values for those three methods.

The results are shown in Figure 5.5. Similarly to the results obtained when working with Naïve Bayes (in Chapter 4), the MR method again shows a much stronger correlation coefficient than HIP, and HRE–TAN also has much higher

r value than HIP. This means that MR and HRE–TAN tend to obtain higher predictive accuracy for the instances that belong to the majority class. By contrast, HIP's difference between Sen. and Spe. is much less correlated with the degree of class imbalance, which shows HIP's ability to predict well both the majority and the minority class.



(A) r(MR+TAN) = 0.798

(B) r(HRE–TAN) = 0.789

(C) r(HIP+TAN) = 0.208

FIGURE 5.5 Values of the Correlation Coefficient between the Degree of Class Imbalance and the Differences between Sen. and Spe. for MR+TAN, HRE–TAN and HIP+TAN

5.4.4 Comparing HIP and MR When Working with TAN

In this section, we compare the experimental results obtained by the HIP and MR methods when they are used to select features for the TAN classifier, in a data pre-processing phase. Table 5.10 only shows the GMean values obtained by HIP and MR methods working with TAN for different datasets; the values of sensitivity and specificity are referred to Sen. and Spe. values in the previous corresponding tables, i.e. Table 5.1 – 5.4 for HIP and Tables 5.5 – 5.8 for MR.

In Table 5.10, the boldface figures denote the higher values of GMean between the two methods. As shown in the table, HIP outperforms MR in 17 out of 28 datasets, while MR outperforms HIP in 11 out of 28 datasets. HIP was particularly successful in the *Drosophila melanogaster* datasets, where HIP outperforms MR in 6 out of 7 datasets; and in the *Saccharomyces cerevisiae* datasets, where HIP outperforms MR in all 7 datasets. MR was more successful in the other datasets.

We also conducted a statistical significance test (i.e. the two-tailed Wilcoxon signed-rank test at 0.05 of significance level) on the GMeans values, and the result reveals that there is no significant difference between HIP and MR when working with TAN.

TABLE 5.10 Predictive Accuracy (GMean Values) for Tree Augmented Naïve Bayes with the Hierarchical HIP and MR Methods

Organism	***Caenorhabditis elegans Datasets***						
GO Types	BP	MF	CC	BP+MF	BP+CC	MF+CC	BP+MF+CC
HIP + TAN	59.4	**46.6**	52.8	62.7	64.0	55.2	**65.5**
MR + TAN	**63.4**	46.5	**53.0**	**66.2**	**66.0**	**56.9**	64.5
Organism	***Drosophila melanogaster Datasets***						
GO Types	BP	MF	CC	BP+MF	BP+CC	MF+CC	BP+MF+CC
HIP + TAN	**64.8**	48.8	**62.8**	**62.0**	**63.3**	**68.0**	**69.3**
MR + TAN	62.0	**58.8**	57.0	61.7	57.8	61.9	68.3
Organism	***Mus musculus Datasets***						
GO Types	BP	MF	CC	BP+MF	BP+CC	MF+CC	BP+MF+CC
HIP + TAN	56.0	**67.9**	**60.3**	58.2	63.2	63.9	61.3
MR + TAN	**60.6**	67.3	57.2	**66.6**	**64.4**	**67.9**	**66.4**
Organism	***Saccharomyces cerevisiae Datasets***						
GO Types	BP	MF	CC	BP+MF	BP+CC	MF+CC	BP+MF+CC
HIP + TAN	**62.4**	**46.0**	**46.2**	**65.5**	**68.2**	**50.3**	**69.8**
MR + TAN	51.1	0.0	44.5	43.2	51.7	31.0	57.3

5.4.5 Scalability of Computational Running Time for Different Feature Selection Methods

We estimated the computational running time for different feature selection methods working with the TAN and TAN classifier without feature selection, following the same approach mentioned in Section 4.7.4.4. Recall that this approach essentially involves estimating the computational time in the largest dataset - *C. elegant* dataset with BP+MF+CC features - and in the smallest dataset - *D. melanogaster* dataset with CC features. The experiments to estimate the algorithms' computational times were run on one iMac equipped with one 2.9 GHz Intel Core i5 CPU, 2×4 GB 1600 MHz DDR3 memory, one Macintosh hard drive and OS X (version 10.8.2) operating system.

Overall, in the experiments with the *Caenorhabditis elegans* dataset, MR+TAN is the most time-consuming algorithm, taking 205.8 hours to run; while CFS+TAN is the least time-consuming algorithm. This result for MR is consistent with the one reported in Section 4.7.4.4 (referring to experiments with Naive Bayes), i.e. MR is the most time-consuming feature selection method in both cases.

Recall that, when working with the lazy learning version of the TAN classifier, MR selects one subset of features that are used for building one Maximum Weight Spanning Tree (MST) for each testing instance, so the total experimental time spent on the whole dataset is substantially higher than the time spent by MR working with the Naive Bayes classifier.

CFS+TAN shows the best efficiency, since it works with the eager learning version of the TAN classifier. More precisely, CFS selects one subset of features, which are used for building one MST for classifying all testing instances. Note that we estimated not only the total computational running time for the hierarchical embedded feature selection method newly proposed in this chapter, i.e. HRE–TAN, but also the time for building the Hierarchical Redundancy Eliminated-Maximum Weight Spanning Tree (HRE–MST), the main procedure of the HRE–TAN algorithm. Comparing with the computational time of other lazy learning-based filter feature selection methods reported in Table 5.11, HRE–TAN is the second most time-consuming method, and the HRE–MST procedure indeed takes a large part (i.e. 91.9%) of the time taken by HRE–TAN.

In the experiments with the much smaller *Drosophila melanogaster* dataset, HRE–TAN is the most time-consuming algorithm, taking 348.6 seconds; while CFS+TAN is the least time-consuming, taking 6.3 seconds.

Hence, the range of computational running time for these experiments varies from 6.3 seconds for the fastest method on the smallest dataset to 8.6 days for the slowest method on the largest dataset.

TABLE 5.11 Estimated Scalability of Computational Time (in Seconds) for Each Feature Selection Method

Datasets	**Algorithms**				
	TAN	HIP + TAN	Ent_{HIP-k} + TAN	Rele_{HIP-k} + TAN	HRE-MST
C. elegans	342,871.6	52,814.2	52,379.1	60,459.6	653,852.2
(BP+MF+CC)	CFS + TAN	MR + TAN	Ent_{MR-k} + TAN	Rele_{MR-k} + TAN	HRE-TAN
	16,686.7	740,896.2	459,824.5	517,679.1	711,681.4
	TAN	HIP + TAN	Ent_{HIP-k} + TAN	Rele_{HIP-k} + TAN	HRE-MST
D. melanogaster	98.9	225.4	11.2	11.8	239.1
(CC)	CFS + TAN	MR + TAN	Ent_{MR-k} + TAN	Rele_{MR-k} + TAN	HRE-TAN
	6.3	236.9	20.1	19.6	348.6

5.5 Rank for HIP-Selected GO Terms Highly-Related with Ageing

As the HIP method was overall the best performing feature selection method when working with TAN, we computed the ranks of GO terms selected by HIP for the BP+MF+CC datasets (the datasets with the largest number of features), for each of the 4 model organisms. The top-ranked terms are shown in Tables

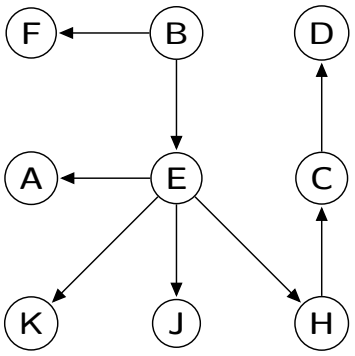


FIGURE 5.6 Example of Built HRE–MST with Node E Having 5 Connections

5.12 – 5.13. Each of the tables consists of 7 columns; the first three columns have self-explanatory names. The rank in column 4 is based on two criteria. The first ranking criterion is the "Frequency of Selection" in column 5, which means the number of times the GO term was selected by HIP for classifying the testing instances. The second, tie-breaking ranking criterion is the "Frequency in Edges" in column 6, which means the number of edges containing the GO term in the trees built by TAN for classifying the test instances. Recall that, for building the tree, each feature is allowed to have at most one parent feature, but each feature could be the parent for more than one child features. For example, as shown in Figure 5.6, node E has the largest number of connections to other nodes (being the child node for node B, and a parent node for other 4 nodes). This type of node could be called a "hub", in the context of this small example graph. The "hub" node plays an important role in the tree. Hence, a feature could act as a "hub" node if that feature is the parent for many nodes. Note that, in terms of the relationship between "Frequency of Selection" and "Frequency in Edges", the value of the latter will always be not smaller than the value of the former, since one selected feature should be included in at least one edge. The class label in the column "Predicted Class" is the most frequent class label in the set of instances with value "yes" ("1") for the corresponding GO term.

Note that Tables 5.12 – 5.13 are different from Table 4.7 shown in Chapter 4 in several ways as follows. First, the GO terms included in Tables 5.12 – 5.13 were

TABLE 5.12 Most Frequently Selected GO Terms by the HIP Method in *Caenorhabditis elegans* and *Drosophila melanogaster* Datasets

GO Term ID	GO Term Type	GO Term Name	Rank	Freq. of Selection	Freq. in Edges	Predicted Class
Caenorhabditis elegans						
GO:0045202	CC	synapse	1	572	2394	Anti
GO:0000003	BP	reproduction	2	572	1929	Anti
GO:0005576	CC	extracellular region	3	572	1095	Anti
GO:0016209	MF	antioxidant activity	4	572	697	Pro
GO:0040007	BP	growth	5	572	633	Pro
GO:0022610	BP	biological adhesion	6	568	1046	Pro
GO:0000988	MF	protein binding transcription factor activity	7	567	801	Pro
GO:0009055	MF	electron carrier activity	8	567	779	Anti
GO:0031974	CC	membrane-enclosed lumen	9	567	769	Anti
GO:0044456	CC	synapse part	10	567	718	Anti
Drosophila melanogaster						
GO:0009055	MF	electron carrier activity	1	130	199	Pro
GO:0005576	CC	extracellular region	2	130	193	Pro
GO:0000003	BP	reproduction	3	130	184	Anti
GO:0044456	CC	synapse part	4	130	174	Pro
GO:0045202	CC	synapse	5	130	152	Pro
GO:0016209	MF	antioxidant activity	6	127	354	Pro
GO:0005198	MF	structural molecule activity	7	127	180	Pro
GO:0030234	MF	enzyme regulator activity	8	126	144	Anti
GO:0004872	MF	receptor activity	9	125	189	Anti
GO:0023052	BP	signaling	10	125	171	Pro

TABLE 5.13 Most Frequently Selected GO Terms by the HIP Method in *Mus musculus* and *Saccharomyces cerevisiae* Datasets

GO Term ID	GO Term Type	GO Term Name	Rank	Freq. of Selection	Freq. in Edges	Predicted Class
Mus musculus						
GO:0044456	CC	synapse part	1	102	354	Anti
GO:0005198	MF	structural molecule activity	2	102	344	Pro
GO:0005576	CC	extracellular region	3	102	270	Pro
GO:0005623	CC	cell	4	102	191	Anti
GO:0045202	CC	synapse	5	102	124	Anti
GO:0030054	CC	cell junction	6	99	248	Anti
GO:0016209	MF	antioxidant activity	7	99	246	Pro
GO:0023052	BP	signaling	8	99	207	Pro
GO:0031012	CC	extracellular matrix	9	99	176	Pro
GO:0022610	BP	biological adhesion	10	99	120	Pro
Saccharomyces cerevisiae						
GO:0005085	MF	guanyl-nucleotide exchange factor activity	1	238	358	Anti
GO:0004872	MF	receptor activity	2	238	282	Anti
GO:0022414	BP	reproductive process	3	234	511	Anti
GO:0009295	CC	nucleoid	4	234	321	Anti
GO:0005933	CC	cellular bud	5	231	479	Anti
GO:0000988	MF	protein binding transcription factor activity	6	231	340	Anti
GO:0005622	CC	intracellular	7	231	283	Anti
GO:0032126	CC	eisosome	8	231	243	Anti
GO:0030234	MF	enzyme regulator activity	9	230	403	Anti
GO:0040007	BP	growth	10	230	277	Anti

selected by the HIP method, whilst the GO terms in Table 4.7 were selected by MR. In addition, in Tables 5.12 – 5.13, the ranking criteria are firstly "selection frequency", and then "frequency in edges", whereas for Table 4.7, apart from the same ranking criterion of "selection frequency", it uses the p-value and relevance value as other types of ranking criteria. The main reasons for adopting different ranking criteria for GO terms is due to the difference in the feature selection strategies used by the two methods. Recall that the HIP method selects the features (GO terms) regardless of their relevance values, whereas the MR method selects features according to their corresponding relevance values. Hence, for Tables 5.12 – 5.13, the relevance and p-value are not used as the ranking criteria for identifying the most relevant GO terms. The other difference between Tables 5.12 – 5.13 and Table 4.7 is that the former ones not only include "biological process" GO terms (like Table 4.7), but also include the other two types of GO terms, i.e. "molecular function" and "cellular component".

As shown in Tables 5.12 – 5.13, several GO terms were selected across three out of the four model organisms: Synapse (GO:0045202), Extracellular Region (GO:0005576), and Antioxidant Activity (GO:0016209) are top-ranked terms in the worm, fly and mouse datasets. Other GO terms were selected across two model organisms: Reproduction (GO:0000003) and Electron Carrier Activity (GO:0009055) are top-ranked in the worm and fly datasets; Protein Binding Transcription Factor Activity (GO:0000988) in the worm and yeast datasets; Receptor Activity (GO:0004872) and Enzyme Regulator Activity (GO:0030234) in the fly and yeast datasets.

Briefly, several of these very often selected GO terms fit well with some ageing-related hypotheses. For example, oxidative processes produce byproducts, i.e. ROS (reactive oxygen species), that can cause damage and crosslink DNA [120]; and antioxidant activity, which can mitigate the harmful effects of high-levels of ROS and is also related to the hypothesis that calorie restriction can delay ageing, was found to be able to extend the longevity of model organisms like worms, mice and flies [106, 107, 121, 130]. As another example, in terms of the link between reproduction and ageing, in *C. elegans*, mutations in the *daf-2* gene reduce insulin/insulin-like growth factor-1 (IGF-1) signaling and lead to extended lifespan and delayed reproduction [68].

Chapter 6

Lazy Hierarchical Feature Selection Methods with Bayesian Network Augmented Naïve Bayes Classifiers

6.1 Introduction

In this chapter, we firstly propose a Bayesian Network Augmented Naïve Bayes (BAN) classifier that exploits background knowledge in the Gene Ontology (GO) to define the network topology. This classifier is called GO–BAN and was firstly described in [122]. We propose two methods for constructing the network topology that is used by the BAN classifier, which is a more complicated type of Semi-naïve Bayesian classifier than TAN. The first method, called *Flat Feature Selection with Gene Ontology-based Bayesian Network Augmented Naïve Bayes (FFS+GO–BAN)*, is used for building the GO–BAN classifier using the Gene Ontology features selected by flat feature selection methods; whereas the second method, called *Hierarchical Feature Selection with Gene Ontology-based Bayesian Network Augmented Naïve Bayes (HFS+GO–BAN)*, is used for building the GO–BAN classifier using the Gene Ontology features selected by hierarchical feature selection methods. In this chapter, in addition to evaluating the performance of

the newly proposed methods, we compare the performance of all proposed hierarchical feature selection methods (in Chapters 4 and 5) combined with different types of Bayesian network classifiers, i.e. NB, TAN and GO–BAN. The BAN network topology construction methods described in this chapter, as well as part of the computational results reported here, have been published in *the 6^{th} ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (ACM–BCB 2015) [124].*

6.2 The Proposed Gene Ontology-Based Bayesian Network Augmented Naïve Bayes (GO–BAN) Classifier

BAN is a type of Semi-naïve Bayes classifier, as discussed in Chapter 2. Unlike NB (where no parent feature is allowed for each feature, and only the class attribute is a parent of all features) and TAN (where one parent feature is allowed for each feature, in addition to the class attribute), a BAN classifier allows each feature to have more than one parent features in the Bayesian network topology. More precisely, as shown in Equation 6.1, in a BAN classifier, the probability of each class value y given the values of the features in the instance is proportional ("$\propto$" symbol) to the prior probability of y multiplied by the product of the conditional probability of each feature x_i given the set of x_i's parent nodes in the network – which includes parent features $Par(x_i)$ and the class y. After computing the probability of each class for the current instance using Equation 6.1, a BAN classifier assigns to the instance the class value with the highest probability.

$$P(y|x_1, x_2, ..., x_n) \propto P(y) \prod_{i=1}^{n} P(x_i|Par(x_i), y) \tag{6.1}$$

In conventional BAN classifiers, the network topology is learnt from the dataset, by assuming the set of features is "flat", i.e. not taking into account hierarchical relationships among features. Here we propose to construct a BAN's network

topology by directly adopting the hierarchical relationships occurring in the available GO data, in order to reduce the computational time needed for building the classifier and exploit the valuable background knowledge encoded in the GO graph that is pre-defined by expert biologists. This type of BAN classifier is here called "Gene Ontology-based BAN" (GO–BAN). That is, in the feature network of the GO–BAN classifier, each feature (GO term f) has a set of parents given by the parents of f in the GO hierarchy, plus the class attribute (which is a parent for all features, like in conventional BAN classifiers).

Figure 6.1 shows an example of network topology for a BAN classifier based on Gene Ontology data. As pre-defined by the GO's hierarchical relationships (represented as solid lines, whereas the dashed lines denote the dependency relationships between an individual feature and the class attribute), term GO:0044765 is the child of terms GO:0006810 and GO:0044699, and the parent of term GO:0045056. This type of hierarchical relationship will be directly used by the GO–BAN classifier, as discussed in the next section.

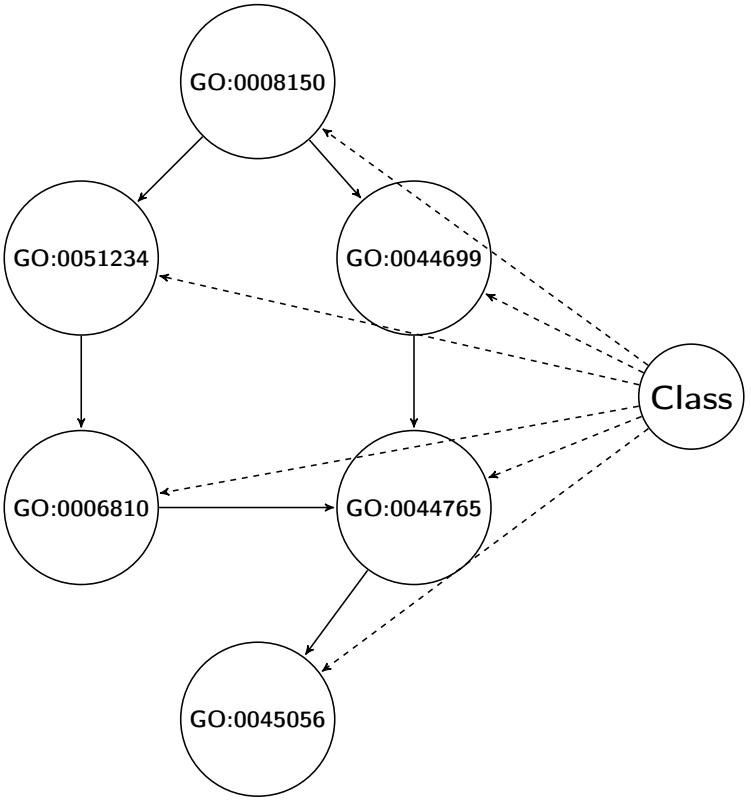


FIGURE 6.1 Example of Topology of a BAN Classifier Based on Gene Ontology Data

6.3 Proposed Methods for Constructing the Network Topology of a GO–BAN Classifier

We propose two methods to construct the BAN network topology containing the Gene Ontology features (GO terms) selected in a data pre-processing phase, where that network topology will be directly used by GO–BAN classifier, as stated in the previous section. The construction of the BAN network topology is not trivial, because the feature selection methods can select features that are hierarchically related (one is the ancestor or descendant of the other), but are not directly connected by an edge in the GO DAG. For instance, in Figure 6.2, a method could select features A and D but not feature B. In such cases, if the BAN network with the selected features contained only edges occurring in the GO DAG, there would be no edge connecting A and D in the BAN, suggesting these features are independent, which would be misleading, given their hierarchical dependency. Therefore, it is necessary to create artificial edges, not present in the GO DAG, which are nonetheless based on hierarchical dependencies represented in the GO DAG, so that these artificial edges can be used in the BAN network.

Hence, we propose two methods for constructing the GO–BAN network based on the features selected in a pre-processing phase and on the structure of the GO DAG. The first BAN network construction method was designed for the case where features have been selected by a flat feature selection method (FFS) (i.e. CFS [48] in this thesis, but other methods could be used). The second BAN network construction method was designed for the case where features have been selected by a hierarchical feature selection method (HIP and MR in this thesis, but again other hierarchical methods could be used).

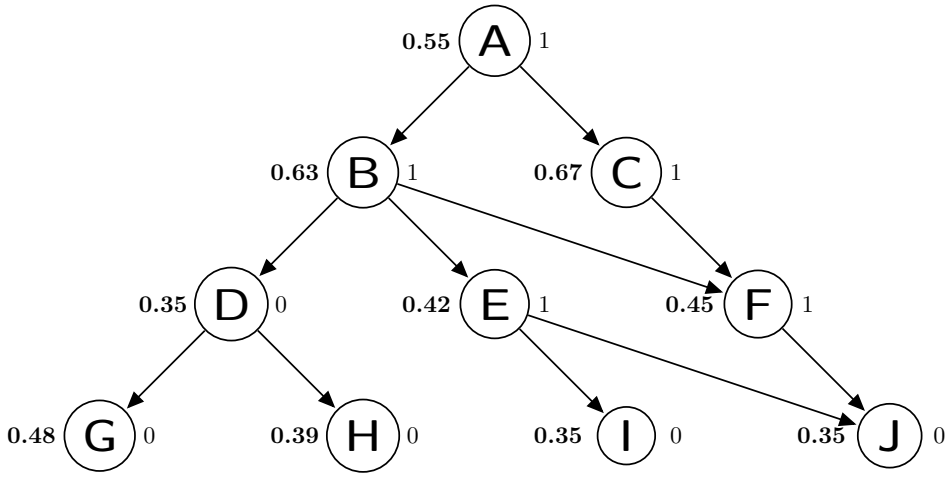


FIGURE 6.2 Example of a Small DAG of Features

6.3.1 Flat Feature Selection with Gene Ontology-Based Bayesian Network Augmented Naïve Bayes (FFS+GO–BAN)

Here we introduce the method (described in Algorithm 6.1) for constructing the GO–BAN classifier using the features selected by a conventional flat feature selection method.

In the first phase of Algorithm 6.1, in lines 1 – 3, the feature DAG, training dataset and testing dataset will be initialised. The initial feature DAG simply contains one node for each GO term (feature) in the dataset and all the edges in the GO DAG where both GO terms connected by the edge are used as features in the dataset. Next, in line 4, the flat feature selection process will be conducted; then the set of selected features $\mathbb{X}_{FFS}$ will be used to re-create the training and testing datasets, in lines 5 – 6.

The second phase (lines 7 – 12) of FFS+GO–BAN (Algorithm 6.1) re-constructs the edges between selected features according to the pre-defined hierarchical relationships in the DAG created in line 1. In details, for each feature x_s selected by FFS, the algorithm considers all paths leading from a root node of the DAG to x_s. As shown in lines 9 – 11, for each of those paths, the algorithm finds the closest ancestor of x_s in that path that was also selected by FFS, denoted (Closest Selected

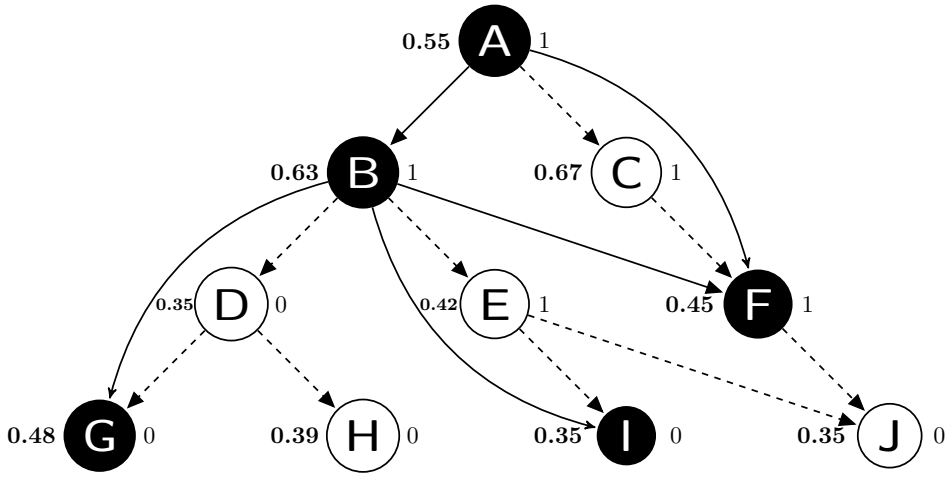


FIGURE 6.3 Example DAG with Nodes Selected by a Flat Feature Selection Method and Corresponding Edges Constructed According to the Gene Ontology Hierarchical Structure Information (FFS+GO–BAN Algorithm)

Ancestor) CloSelAnc(x_s), and adds CloSelAnc(x_s) to the set of parents of x_s in the GO–BAN network. That is, it adds an edge pointing from CloSelAnc(x_s) to x_s on the GO–BAN network. In the third and last phase of Algorithm 6.1, lines 13 – 15, each testing instance is classified using the previously constructed GO–BAN network.

To further explain how Algorithm 6.1 works, Figure 6.3 shows an example DAG where the selected nodes (features) are shown in black and the edges represent generalisation-specialisation relationships among GO terms (features) in the GO DAG. The dashed edges are the edges that are included in the GO DAG but are not included in the constructed GO–BAN network. The solid edges are the edges included in the constructed GO–BAN network; some of these solid edges represent parent-child relationships between selected features in the GO DAG, whilst other solid edges represent new edges which were artificially created to represent a direct connection between two selected features, which are separated by two or more edges in a given path of the GO DAG. Note that a selected node can have more than one selected ancestor nodes in an individual path, e.g. node G has two selected ancestors, B and A. In this case only its closest selected ancestor node B – in the path A–B–D–G – will be assigned to the set of parent nodes of G in lines 9 – 11 of Algorithm 6.1. Analogously, only the closest selected ancestor of node I in the path A–B–E–I, namely node B, will be added to the set of parents

of node I. Furthermore, node F is assigned two parent nodes, namely B, which is F's closest selected ancestor in path A–B–F, and A, which is F's only selected ancestor in path A–C–F.

Algorithm 6.1 Flat Feature Selection with Gene Ontology-Based Bayesian Network Augmented Naïve Bayes (FFS+GO–BAN)

Initialise **DAG** with all features in Dataset;
Initialise **TrainSet**;
Initialise **TestSet**;
$\mathbb{X}_{FFS}$ = **FFS**(**TrainSet**);
Create **TrainSet_FFS** with features $\mathbb{X}_{FFS}$;
Create **TestSet_FFS** with features $\mathbb{X}_{FFS}$;
for each $x_s \in \mathbb{X}_{FFS}$ **do**
    Par$(x_s) = \emptyset$;
    for each path k in **DAG** from root to x_s **do**
        Par$(x_s) \leftarrow$ **Par**$(x_s) \cup$ **CloSelAnc**(x_s);
    end for
end for
for each **Inst_FFS**$_{<w>} \in$ **TestSet_FFS do**
    Classify(**Par**$(\mathbb{X}_{FFS})$, **TrainSet_FFS**, **Inst_FFS**$_{<w>}$);
end for

Note that flat feature selection (FFS) methods cannot guarantee the elimination of hierarchical redundancies between features. Therefore, FFS methods can select features that have the same value (either "1" or "0") in an instance and are located in the same path in the GO DAG. In the example DAG in Figure 6.2, the FFS method has selected features A and B, which is a case of hierarchical redundancy (the value "1" of B in an instance implies the value "1" of A in that instance). Such hierarchical redundancies in the GO–BAN network are avoided by using hierarchical feature selection methods, as discussed in the next Section.

6.3.2 Hierarchical Feature Selection with Gene Ontology-Based Bayesian Network Augmented Naïve Bayes (HFS+GO–BAN)

Recall that the Hierarchical Feature Selection (HFS) methods used in this work perform lazy learning, i.e. they select a set of features specific for each testing instance. We evaluate the predictive performance of GO–BAN when using two lazy HFS methods in a pre-processing phase, i.e. HIP and MR (described in Chapter 4). Hence, in this Section we propose another method to construct the GO–BAN network topology from the set of features selected by HIP or MR. Note that the proposed method is generic enough to be used with any other lazy HFS method, which can eliminate the hierarchical redundancy.

Algorithm 6.2 works in a way analogous to Algorithm 6.1. The core part of both algorithms consists of finding the closest selected ancestor of each selected feature x_s in each path of the GO DAG and adding that ancestor to the set of parents of feature x_s. The main difference between these two algorithms is as follows. Since Algorithm 6.1 uses an eager feature selection algorithm, its core part (the loop in lines 7 – 12) is performed before processing the testing instances in lines 13 – 15. By contrast, since Algorithm 6.2 uses a lazy feature selection method, both the use of a HFS method in line 5 and the algorithm's core part (the loop in lines 8 – 13) are performed within a loop over all testing instances. Another difference is that line 10 of Algorithm 6.1 involves finding the closest selected ancestor of selected feature x_s in path k; whilst the corresponding line

11 of Algorithm 6.2 is somewhat simpler; it is not necessary to select the closest ancestor of x_s among several ancestors, simply because x_s will have at most one selected ancestor feature. This is due to the fact that the HFS method executed in line 5 (i.e. HIP or MR) eliminates hierarchical redundancies among features.

The initialisation phase of HFS+GO–BAN (lines 1 – 3 in Algorithm 6.2) is the same as the initialisation phase of Algorithm 6.1. Then, for each testing instance ($\mathbf{Inst}_{<w>}$), a lazy learning HFS method (either HIP or MR) will be run (line 5 in Algorithm 6.2). Next, the set of hierarchically selected features $\mathbb{X}_{HFS}$ is used to re-create the new training dataset **TrainSet_HSF** and the current testing instance $\mathbf{Inst\_HSF}_{<w>}$. In lines 8 – 13, the GO–BAN network is constructed. For each selected feature x_s in $\mathbb{X}_{HSF}$, for each path in the DAG from a root node to x_s, the only selected ancestor of x_s (if such ancestor exists) is added to the set of parents of x_s in the GO–BAN network in line 11.

To further explain how Algorithm 6.2 works when HIP is used, consider the example DAG in Figure 6.4, where the nodes selected by HIP are marked in black (nodes D, E, I, F and J). Each of these nodes has at most one selected ancestor node in each path from the root to that node. Hence, Algorithm 6.2 assigns node E as the parent of node I in path A–B–E–I; node E as the parent of node J in path A–B–E–J; node F as the parent of node J in paths A–B–F–J and A–C–F–J. Nodes D, E, F are not assigned any parent, since none of their ancestor nodes in the DAG were selected by HIP.

To further explain how Algorithm 6.2 works when MR is used, consider the DAG in Figure 6.5, where again the selected nodes are marked in black (nodes B, G, H, C, I and J). Again, each selected node has at most one selected ancestor node in each path from the root to that node. Hence, Algorithm 6.2 assigns node B as the parent of node G in path A–B–D–G; node B as parent of node H in paths A–B–D–H and A–B–H; node B as parent of node I in path A–B–E–I; node B as parent of node J in paths A–B–E–J and A–B–F–J; node C as parent of node J in path A–C–F–J.

Algorithm 6.2 Hierarchical Feature Selection with Gene Ontology-Based Bayesian Network Augmented Naïve Bayes (HFS+GO–BAN)

```
1: Initialise DAG with all features in Dataset;
2: Initialise TrainSet;
3: Initialise TestSet;
4: for each Inst_<w> ∈ TestSet do
5:     𝕏_HFS = HFS(DAG, TrainSet, Inst_<w>);
6:     Create TrainSet_HFS with features 𝕏_HFS;
7:     Create Inst_HFS_<w> with features 𝕏_HFS;
8:     for each x_s ∈ 𝕏_HFS do
9:         Par(x_s) = ∅;
10:        for each path k in DAG from root to x_s do
11:            Par(x_s) ← Par(x_s) ∪ SelAnc(x_s);
12:        end for
13:    end for
14:    Classify(Par(𝕏_HFS), TrainSet_HFS, Inst_HFS_<w>);
15: end for
```

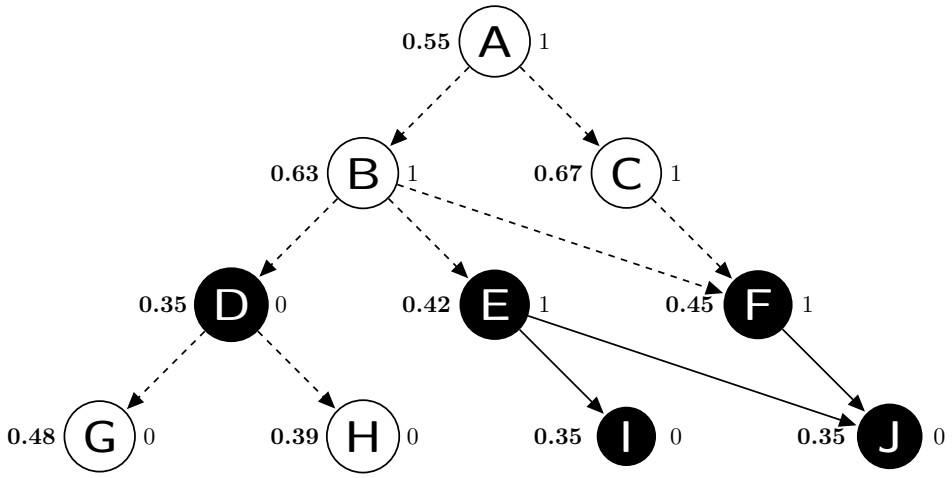


FIGURE 6.4 Example DAG with Nodes Selected by HIP and Corresponding BAN Network Constructed according to the Gene Ontology Hierarchy (HIP+GO–BAN Algorithm)

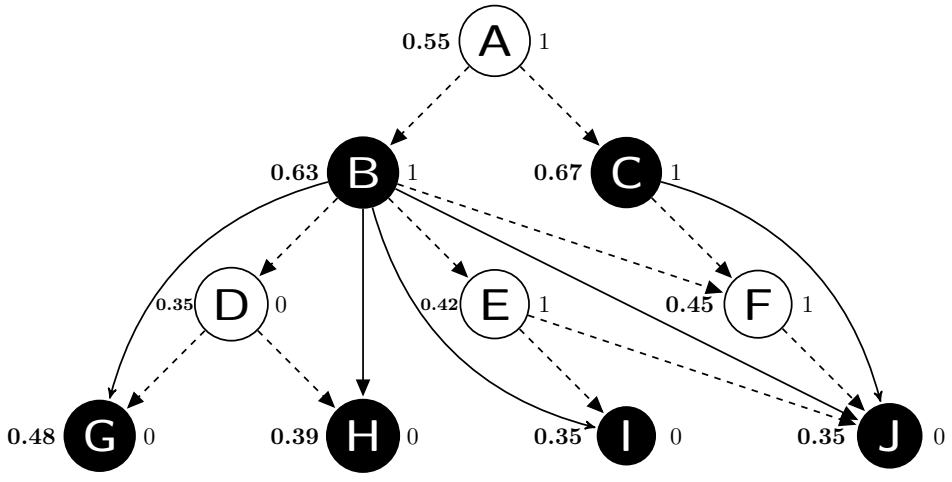


FIGURE 6.5 Example DAG with Nodes Selected by MR and Corresponding Network Constructed according to the Gene Ontology Hierarchy (MR+GO–BAN Algorithm)

6.4 Computational Experiments

6.4.1 Experimental Methodology

We used the ageing-related datasets which have already been adopted in Chapters 4 and 5, i.e. 28 datasets that consist of three types of GO terms (BP, MF, CC), and their different types of combination (BP+MF, BP+CC, MF+CC and BP+MF+CC).

In the experiments reported in this section, there are 4 methods being compared, namely: GO–BAN without feature selection (as a baseline method), GO–BAN based on features selected by the HIP method (HIP+GO–BAN), GO–BAN based on features selected by the MR method (MR+GO–BAN), and GO–BAN based on features selected by the CFS method (a type of flat feature selection method). We also used the well-known 10-fold cross validation procedure to evaluate the performance of classifiers as measured by their GMean value, as discussed in Chapter 4.

6.4.2 Experimental Results

Tables 6.1 – 6.4 compare the predictive performance of the three above mentioned feature selection methods working with GO–BAN and GO–BAN without feature selection. Each table contains results for a different model organism. In these tables, recall that GM stands for the geometric mean of sensitivity and specicity, defined as $\mathbf{GMean} = \sqrt{Sen. \times Spe.}$, where *Sen.* is the proportion of pro-longevity instances corrently predicted as pro-longevity and *Spe.* is the proportion of anti-longevity instances correctly predicted as anti-longevity. In general, considering the results in all 4 tables (Tables 6.1 – 6.4), HIP+GO–BAN shows the best performance among all 4 methods, being ranked as the best method in 23 (out of 28) datasets (GMean values in boldface). In terms of the average ranks for those methods, HIP+GO–BAN obtains the best rank of 1.2 on average over the 28 datasets, which is better than the average rank of MR+GO–BAN (2.2), CFS+GO–BAN (2.8) and GO–BAN with no feature selection (3.8).

TABLE 6.1 Predictive Accuracy for GO–BAN with Hierarchical HIP and MR, and Flat CFS Method in *Caenorhabditis elegans* Datasets

Caenorhabditis elegans Datasets												
Feature Types	**GO–BAN without Feature Selection**			**Hier. HIP + GO–BAN**			**Hier. MR + GO–BAN**			**Flat CFS + GO–BAN**		
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	28.7 ± 2.2	86.5 ± 1.8	49.8	54.5 ± 3.2	73.4 ± 2.7	**63.2**	52.2 ± 3.1	74.0 ± 2.2	62.2	45.0 ± 2.6	80.9 ± 2.5	60.3
MF	34.7 ± 4.5	66.5 ± 4.5	**48.0**	43.8 ± 4.5	52.5 ± 5.2	**48.0**	35.5 ± 3.0	63.3 ± 3.4	47.4	31.4 ± 6.6	70.9 ± 6.0	47.2
CC	33.7 ± 4.5	81.4 ± 2.2	52.4	55.1 ± 5.0	63.5 ± 4.0	**59.2**	40.8 ± 4.3	73.1 ± 2.6	54.6	35.7 ± 4.3	74.4 ± 3.9	51.5
BP+MF	30.0 ± 2.7	84.7 ± 1.7	50.4	55.9 ± 3.2	74.1 ± 2.5	64.4	63.8 ± 2.2	73.2 ± 2.1	**68.3**	52.1 ± 3.7	77.6 ± 2.2	63.6
BP+CC	29.1 ± 2.1	86.6 ± 1.7	50.2	58.7 ± 3.6	72.7 ± 2.5	**65.3**	54.0 ± 2.8	74.7 ± 2.3	63.5	47.4 ± 2.7	79.1 ± 1.5	61.2
MF+CC	35.3 ± 2.9	80.2 ± 3.2	53.2	55.9 ± 3.1	64.5 ± 3.6	**60.0**	47.1 ± 3.4	70.2 ± 3.9	57.5	46.5 ± 4.1	72.1 ± 4.0	57.9
BP+MF+CC	31.2 ± 2.9	85.2 ± 1.5	51.6	58.1 ± 3.8	73.4 ± 2.6	**65.3**	55.3 ± 4.0	72.0 ± 2.6	63.1	50.7 ± 4.1	75.4 ± 2.1	61.8

Table 6.2 Predictive Accuracy for GO–BAN with Hierarchical HIP and MR, and Flat CFS Method in *Drosophila melanogaster* Datasets

***Drosophila melanogaster* Datasets**												
Feature Types	**GO–BAN without Feature Selection**			**Hier. HIP + GO–BAN**			**Hier. MR + GO–BAN**			**Flat CFS + GO–BAN**		
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	100.0 ± 0.0	0.0 ± 0.0	0.0	75.8 ± 4.4	52.8 ± 8.6	**63.3**	80.2 ± 3.5	44.4 ± 10.2	59.7	78.0 ± 4.1	25.0 ± 7.8	44.2
MF	91.2 ± 3.3	26.5 ± 3.4	49.2	64.7 ± 7.2	50.0 ± 10.0	56.9	80.9 ± 5.2	47.1 ± 9.1	**61.7**	85.3 ± 4.3	32.4 ± 7.1	52.6
CC	93.5 ± 2.6	28.6 ± 11.1	51.7	79.0 ± 6.6	46.4 ± 11.4	60.5	85.5 ± 4.6	42.9 ± 10.2	60.6	88.7 ± 3.5	46.4 ± 11.4	**64.2**
BP+MF	97.8 ± 1.5	0.0 ± 0.0	0.0	72.8 ± 3.9	63.2 ± 9.3	**67.8**	80.4 ± 3.7	44.7 ± 8.2	59.9	83.7 ± 3.5	28.9 ± 6.2	49.2
BP+CC	98.9 ± 1.1	0.0 ± 0.0	0.0	73.6 ± 4.7	62.2 ± 8.4	**67.7**	80.2 ± 4.1	51.4 ± 10.9	64.2	82.4 ± 4.4	40.5 ± 10.2	57.8
MF+CC	95.3 ± 1.9	31.6 ± 5.3	54.9	80.0 ± 6.2	60.5 ± 7.6	**69.6**	83.5 ± 4.9	55.3 ± 8.2	68.0	90.6 ± 3.0	52.6 ± 4.5	69.0
BP+MF+CC	98.9 ± 1.1	2.6 ± 2.5	16.0	73.9 ± 4.7	68.4 ± 5.3	71.1	81.5 ± 3.7	63.2 ± 7.7	**71.8**	88.0 ± 2.6	44.7 ± 8.2	62.7

Table 6.3 Predictive Accuracy for GO–BAN with Hierarchical HIP and MR, and Flat CFS Method in *Mus musculus* Datasets

***Mus musculus* Datasets**												
Feature Types	**GO–BAN without Feature Selection**			**Hier. HIP + GO–BAN**			**Hier. MR + GO–BAN**			**Flat CFS + GO–BAN**		
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	98.5 ± 1.4	26.5 ± 5.0	51.1	75.0 ± 5.1	70.6 ± 5.1	**72.8**	88.2 ± 4.7	44.1 ± 7.7	62.4	85.3 ± 4.3	44.1 ± 5.9	61.3
MF	90.8 ± 3.3	27.3 ± 10.0	49.8	84.6 ± 3.0	45.5 ± 12.2	**62.0**	87.7 ± 3.0	39.4 ± 10.6	58.8	87.7 ± 2.9	30.3 ± 9.6	51.5
CC	86.4 ± 3.3	35.3 ± 11.2	55.2	80.3 ± 3.0	50.0 ± 10.1	**63.4**	78.8 ± 3.8	44.1 ± 11.1	58.9	78.8 ± 3.3	38.2 ± 12.6	54.9
BP+MF	98.5 ± 1.4	29.4 ± 6.4	53.8	69.1 ± 5.8	70.6 ± 8.1	**69.8**	86.8 ± 4.0	41.2 ± 9.6	59.8	89.7 ± 2.2	41.2 ± 8.0	60.8
BP+CC	98.5 ± 1.4	29.4 ± 6.4	53.8	66.2 ± 6.0	76.5 ± 8.0	**71.2**	77.9 ± 5.3	52.9 ± 9.6	64.2	82.4 ± 5.6	47.1 ± 11.7	62.3
MF+CC	91.2 ± 3.2	26.5 ± 8.8	49.2	79.4 ± 4.2	61.8 ± 12.5	70.0	83.8 ± 5.0	58.8 ± 13.1	**70.2**	79.4 ± 4.8	44.1 ± 9.6	59.2
BP+MF+CC	98.5 ± 1.4	26.5 ± 10.5	51.1	70.6 ± 6.0	76.5 ± 8.8	**73.5**	86.8 ± 4.0	50.0 ± 6.9	65.9	83.8 ± 3.3	52.9 ± 8.4	66.6

TABLE 6.4 Predictive Accuracy for GO–BAN with Hierarchical HIP and MR, and Flat CFS Method in *Saccharomyces cerevisiae* Datasets

Saccharomyces cerevisiae Datasets												
Feature Types	**GO–BAN without Feature Selection**			**Hier. HIP + GO–BAN**			**Hier. MR + GO–BAN**			**Flat CFS + GO–BAN**		
	Sen.	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*	*Sen.*	*Spe.*	*GM*
BP	0.0 ± 0.0	100.0 ± 0.0	0.0	63.3 ± 6.0	76.8 ± 3.1	**69.7**	33.3 ± 8.6	89.7 ± 2.5	54.7	20.0 ± 5.4	94.6 ± 1.9	43.5
MF	0.0 ± 0.0	99.2 ± 0.8	0.0	23.1 ± 6.7	80.2 ± 3.9	**43.0**	0.0 ± 0.0	90.8 ± 3.0	0.0	0.0 ± 0.0	94.7 ± 1.6	0.0
CC	12.5 ± 6.1	99.2 ± 0.8	35.2	29.2 ± 10.2	83.7 ± 4.1	**49.4**	20.8 ± 6.9	93.5 ± 2.7	44.1	20.8 ± 7.5	93.5 ± 1.6	44.1
BP+MF	0.0 ± 0.0	100.0 ± 0.0	0.0	73.3 ± 6.7	71.9 ± 3.0	**72.6**	23.3 ± 7.1	89.6 ± 2.6	45.7	26.7 ± 8.3	96.4 ± 1.1	50.7
BP+CC	0.0 ± 0.0	100.0 ± 0.0	0.0	63.3 ± 10.5	78.4 ± 2.9	**70.4**	40.0 ± 8.3	87.3 ± 2.5	59.1	26.7 ± 6.7	96.6 ± 1.1	50.8
MF+CC	0.0 ± 0.0	100.0 ± 0.0	0.0	41.4 ± 8.3	80.7 ± 3.0	**57.8**	13.8 ± 6.3	88.8 ± 2.3	35.0	13.8 ± 6.3	93.4 ± 1.5	35.9
BP+MF+CC	0.0 ± 0.0	100.0 ± 0.0	0.0	76.7 ± 7.1	73.6 ± 2.8	**75.1**	33.3 ± 5.0	87.0 ± 2.5	53.8	23.3 ± 8.7	94.2 ± 1.6	46.8

More precisely, in terms of results for each type of model organism, the main findings are as follows. In Table 6.1, for the datasets about *Caenorhabditis elegans*, the "anti-longevity" class is the majority class, and overall the values of Spe. are greater than Sen. HIP+TAN obtains the highest GMean value 6 out of 7 times (including one draw with GO–BAN), whilst MR+GO–BAN and GO–BAN obtain only once the highest GMean value.

In Table 6.2, for the datasets about *Drosophila melanogaster*, the "pro-longevity" class is the majority class, and the values of sensitivity are greater than the values of specificity obtained by all algorithms. HIP+GO–BAN obtains 4 out of 7 times the highest GMean value, while MR+GO–BAN obtains 2 times the highest GMean value, and CFS+GO–BAN obtains only once the highest value.

In Table 6.3, for the datasets about *Mus musculus*, "pro-longevity" is the majority class, and the values of sensitivity are greater than the value of specificity obtained by most algorithms overall (with exceptions for HIP+GO–BAN working on BP+MF, BP+CC and BP+MF+CC datasets). HIP+GO–BAN obtains 6 out of 7 times the highest GMean value, while MR+GO–BAN obtains once the highest GMean value.

In Table 6.4, for the datasets about *Saccharomyces cerevisiae*, the "anti-longevity" class is the majority class, and the values of specificity are greater than the values of sensitivity obtained by most algorithms (with exception of HIP+GO–BAN working on BP+MF, BP+MF+CC datasets). Among those algorithms, HIP+GO–BAN obtains all 7 out of 7 times the highest GMean value.

Hence, in all 4 types of datasets, for all model organisms, when comparing the values of *Sen.* and *Spe.*, the highest value is obtained for the measure asscociated with the majority class. That is, as expected, it seems easier to predict the majority class.

We performed a statistical significance test on the predictive accuracies of different feature selection methods by adopting the Friedman test and the Holm *post-hoc* method. As discussed in Chapter 4, the Friedman test is a non-parametric statistical test based on the ranks of each classifier's predictive performance on each dataset [29, 58], and the Holm *post-hoc* method is used for coping with the multiple-comparison problem that arises when applying significance tests to multiple pairwise method comparisons [28]. We used HIP+GO–BAN as the control (best) feature selection method to be compared with the other methods.

TABLE 6.5 Statistical Significance Test Results of the Algorithms' GMean Values According to the Non-Parametric Friedman Test with the Holm *Post-Hoc* Test at the $\alpha = 0.05$ Significance Level

Algorithms	*Ave. Rank*	*P-value*	*Adjusted* α
HIP+GO–BAN (ctrl)	1.2	–	–
MR+GO–BAN	2.2	**3.74 E-03**	0.0500
CFS+GO–BAN	2.8	**3.52 E-06**	0.0250
No FS+GO–BAN	3.8	**4.85 E-14**	0.0167

The detailed results of these significance tests are shown in Table 6.5, where the second column shows the average rank of each method (recall that the lower the rank, the better the predictive performance); the third column shows the calculated p-value; the fourth column shows the adjusted significance level (α). In the third column, a boldfaced value indicates that the p-value is lower than the corresponding adjusted significance level, which means the difference of GMean values between HIP+GO–BAN and the corresponding method is statistically significant. The outcomes of the statistical significance tests show that HIP+GO–BAN significantly outperforms MR+GO–BAN, CFS+GO–BAN and GO–BAN without feature selection.

6.5 Discussion

6.5.1 The Average Dimensionalities of Conditional Probability Tables Created by Different Algorithms

Table 6.6 reports a number of statistics about the size of the constructed GO–BAN's DAGs, when using different feature selection methods. More precisely, the columns referring to GO–BAN without feature selection report the original number of features (**F**) and edges (**E**) in the feature DAG for each dataset, and the average dimensionality of a conditional probability table (CPT) in that DAG, denoted $\mathbf{D}(CPT)$. To calculate $\mathbf{D}(CPT)$, note that each node is associated with a number

of variables given by its number of parent feature nodes plus two – accounting for one class variable (which is a parent of all feature nodes) and the feature represented by the node itself. Since all (feature and class) variables can take two values, the dimensionality of each CPT is given by Equation 6.2, where $\#Par$ is the number of parent features. The table columns referring to GO–BAN using HIP and MR as feature selection methods report the average number of selected features (**AvF**), the average number of edges in the constructed DAG (**AvE**), and the average CPT dimensionality in the DAG for the corresponding feature selection method, where each average is computed over the DAGs constructed for all testing instances (since HIP and MR select a specific feature set for each testing instance) across all 10 cross-validation iterations. Finally, in the table columns referring to GO–BAN using the feature selection method CFS, the average is computed over the 10 cross-validation iterations only, since in each iteration CFS selects the same set of features to classify all available testing instances.

$$\mathbf{D}(CPT) = 2^{(\#Par+2)} \tag{6.2}$$

In general, the three feature selection methods selected substantially fewer features and so the corresponding constructed GO–BAN DAGs had substantially fewer edges, compared with the original DAGs (without performing feature selection). More precisely, among the three feature selection methods, CFS selected the smallest number of features in 27 out of the 28 datasets (the only exception is the dataset for *S. cerevisiae* with MF features). MR selected the largest number of features in all 28 datasets; and the number of features selected by HIP is in general an intermediate value between the numbers selected by the other two methods. However, HIP+GO–BAN constructed DAGs having in general fewer edges than the DAGs constructed by MR+GO–BAN and CFS+GO–BAN.

Figure 6.6 shows the average CPT dimensionality ($\mathbf{D}(CPT)$) in the DAGs constructed by each method, where the average was computed over all the 28 datasets. As shown in this figure, despite CFS selecting a smaller feature set than HIP and MR, the CFS+GO–BAN method constructs DAGs with the largest average CPT dimensionality ($\mathbf{D}(CPT)$) value of 5.65, among the three feature

selection methods – although this value is still much smaller than the value for GO–BAN without feature selection (14.6). This $\mathbf{D}(CPT)$ value of 5.65 for CFS+GO–BAN is substantially higher than the $\mathbf{D}(CPT)$ values obtained by MR+GO–BAN (4.78) and by HIP+GO–BAN (4.26). This indicates that, although CFS selected the smallest number of features, on average the features selected by CFS have a higher number of parent nodes in the constructed DAGs, leading to the highest $\mathbf{D}(CPT)$ values for CFS among all the feature selection methods.

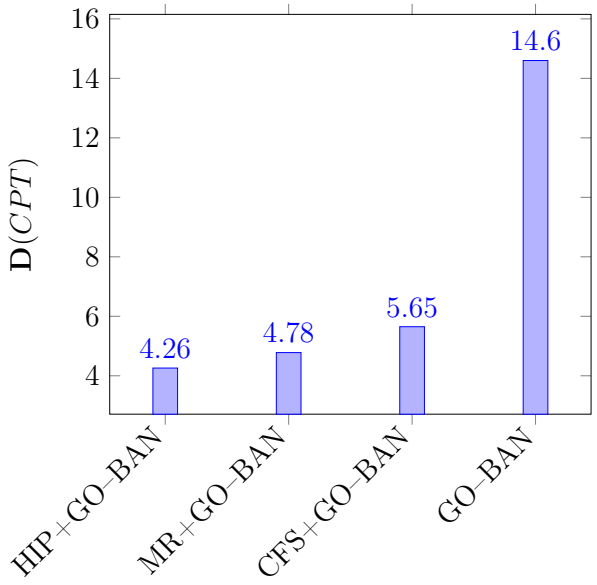


FIGURE 6.6 Average $\mathbf{D}(CPT)$ Values for Different Feature Selection Methods Working with GO–BAN over 28 Datasets

These results are consistent with the discussion in Section 6.3.1, i.e. the features selected by CFS can have more than one ancestor features that have the same values and are also located in the same path in the DAG, constituting a case of hierarchical redundancy (defined in Section 2.7), a type of redundancy that is not eliminated by CFS; and this leads to a higher number of parents per node and so a substantially higher $\mathbf{D}(CPT)$ value for CFS.

Unlike CFS, both HIP and MR remove the hierarchical redundancy between features, which means there will exist at most two nodes being selected and at most one dependency being constructed for each individual path; and this leads to substantially lower $\mathbf{D}(CPT)$ values for HIP+GO–BAN and MR+GO–BAN, by comparison with CFS+GO–BAN.

The reason for HIP+GO–BAN having a smaller $\mathbf{D}(CPT)$ value than MR+GO–BAN is that HIP selected in general substantially fewer features than MR (as

shown in Table 6.6), which led to substantially smaller numbers of edges and parent features per node. In particular, the lowest $\mathbf{D}(CPT)$ value of 4.26 obtained by HIP+GO–BAN suggests that most nodes in the constructed DAG have no parent feature, since a $\mathbf{D}(CPT)$ value of 4 means a CPT has only four probability values, arising from the four combinations of two values of the current feature and two values of the class variable. The small size of the CPTs constructed by HIP+GO–BAN suggests that this method is the one that most mitigates the problem of over-fitting associated with large CPTs. This is because the larger the average dimensionality of CPTs in a constructed DAG, the larger the number of "parameters" (probability values) to be estimated from the training data, and the larger the risk of over-fitting.

TABLE 6.6 Number of Selected Features **F**, Number of Edges **E** and Dimensionalities of CPT Tables **D**(CPT) for the Constructed GO–BAN Classifier

Feature Types	**GO–BAN without FS**			**Hier. HIP + GO–BAN**			**Hier. MR + GO–BAN**			**Flat CFS + GO–BAN**		
Caenorhabditis elegans Datasets												
	F	**E**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)
BP	830	1437	17.66	69.27	2.19	4.13	145.67	32.21	4.95	42.1	8.5	4.83
MF	218	259	10.32	29.81	2.91	4.40	50.52	9.02	4.73	27.8	6.3	4.91
CC	143	217	14.03	29.73	2.09	4.31	54.98	7.84	4.61	23.3	2.9	4.50
BP+MF	1049	1696	16.13	91.88	4.43	4.20	195.41	31.89	4.69	54.4	10.0	4.74
BP+CC	974	1654	17.12	90.01	3.11	4.15	189.84	32.43	4.73	53.9	11.4	4.88
MF+CC	362	476	11.79	51.85	3.89	4.31	102.00	14.57	4.60	40.0	7.5	4.75
BP+MF+CC	1193	1913	15.88	112.96	5.33	4.19	244.66	38.32	4.66	60.9	10.8	4.72
Drosophila melanogaster Datasets												
	F	**E**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)
BP	698	1190	17.28	82.53	3.94	4.21	141.74	19.83	4.66	31.2	5.4	4.77
MF	130	151	10.29	22.87	2.65	4.49	31.76	5.99	4.80	13.3	2.7	4.81
CC	75	101	12.05	20.73	1.58	4.31	27.60	8.39	5.33	14.6	4.6	5.37
BP+MF	829	1341	16.17	120.99	6.39	4.26	172.68	27.38	4.73	31.8	6.4	4.93
BP+CC	774	1291	16.76	100.38	5.02	4.21	167.14	29.84	4.83	33.5	6.6	4.84
MF+CC	206	252	10.94	40.65	3.77	4.38	58.59	10.07	4.73	21.3	5.5	5.07
BP+MF+CC	905	1442	15.83	121.34	7.48	4.22	201.47	31.71	4.97	33.6	7.9	5.08
Mus musculus Datasets												
	F	**E**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)
BP	1039	1836	17.18	128.60	7.48	4.25	197.48	28.37	4.64	36.6	6.5	4.79
MF	182	205	9.68	44.06	4.39	4.41	50.37	10.95	4.92	25.3	8.5	5.47
CC	117	160	12.37	36.68	2.87	4.33	38.75	11.85	5.50	15.7	2.4	4.64
BP+MF	1222	2041	16.06	171.32	11.70	4.29	245.42	38.58	4.69	43.7	10.2	5.04
BP+CC	1157	1996	16.69	164.83	10.29	4.27	234.87	40.58	4.77	40.2	8.4	4.94
MF+CC	300	365	10.74	78.96	7.03	4.37	90.04	19.76	4.99	27.5	7.8	5.24
BP+MF+CC	1340	2201	15.73	207.56	14.51	4.29	286.44	49.50	4.77	46.3	8.9	4.84
Saccharomyces cerevisiae Datasets												
	F	**E**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)	**AvF**	**AvE**	**D**(CPT)
BP	679	1223	18.85	54.58	1.97	4.15	107.24	13.51	4.55	31.4	19.0	7.68
MF	175	209	10.43	24.59	1.78	4.30	40.98	5.90	4.58	35.6	8.4	4.96
CC	107	168	14.56	28.56	1.14	4.16	35.34	9.51	5.15	20.7	18.0	7.98
BP+MF	855	1432	17.12	76.54	3.41	4.19	150.73	17.36	4.51	31.1	18.6	7.69
BP+CC	787	1391	18.26	77.91	2.63	4.14	144.09	21.46	4.65	34.5	33.3	10.57
MF+CC	283	377	12.00	48.11	2.28	4.19	84.59	11.81	4.58	29.8	18.3	7.07
BP+MF+CC	963	1600	16.83	99.96	4.03	4.17	191.24	25.35	4.57	34.9	28.4	9.21

6.5.2 Scalability of Computational Running Time for Different Feature Selection Methods

We estimated the computational time of algorithms by using the iMac equipped with one 2.9 GHz Intel Core i5 CPU, 2×4 GB 1600 MHz DDR3 memory, one Macintosh hard drive and OS X (version 10.8.2) operating system.

Table 6.7 reports the estimated computational running time for different feature selection methods working with the GO–BAN classifier and the GO–BAN classifier without feature selection, following the same approach explained in Section 4.7.4.4. Recall that this approach essentially involves estimating the computational running time in the largest dataset - *C. elegans* with BP+MF+CC features, and in the smallest dataset - *D. melanogaster* with CC features.

Overall, in the experiments with the *Caenorhabditis elegans* dataset, GO–BAN without feature selection is the most time-consuming algorithm, while all other feature selection methods give a significant contribution to reducing the computational running time of the original GO–BAN classifier. The reason is due to the large dimensionalities of the CPTs created by the GO–BAN classifier, i.e. each feature is associated with a CPT having a dimensionality of 15.9, on average. This large average dimensionality leads to a much higher computational time, comparing with the average CPT dimensionality for the other algorithms, i.e. 4.2 for HIP+GO–BAN, 4.7 for MR+GO–BAN and 4.7 for CFS+GO–BAN. Among those three feature selection methods combined with GO–BAN, MR+GO–BAN is still the most time-consuming algorithm, which is consistent with the results for MR working with Naïve Bayes and TAN classifiers reported in Section 4.7.4.4 and 5.4.5, respectively.

In the experiments with the *Drosophila melanogaster* dataset, GO–BAN without any feature selection method is the least time-consuming algorithm, while MR+GO–BAN is the most time-consuming algorithm. The reason why GO–BAN performs fastest is that, in this small dataset, GO–BAN is already fast without feature selection; and the time taken by the feature selection methods is much larger than the time taken by GO–BAN without feature selection.

Overall, the range of computational running time for the experiments varies from 5.6 seconds for the fastest method on the smallest dataset to 14.5 hours for

the slowest method on the largest dataset.

TABLE 6.7 Estimated Scalability of Computational Time (in Seconds) for Each GO–BAN Algorithm

Dataset	GO–BAN	HIP + GO–BAN	MR + GO–BAN	CFS + GO–BAN
C. elegans (BP+MF+CC)	52,295.0	2,073.4	31,703.0	17,739.1
D. melanogaster (CC)	5.6	218.4	234.5	216.0

6.6 Comparison between All Proposed Feature Selection Methods Working with Three Different Types of Bayesian Network Classifiers

In general, the proposed hierarchical feature selection methods show an improvement on the predictive performance of different types of Bayesian Network classifiers, i.e. NB, TAN and GO–BAN in this work. In order to further quantify these improvements, here we compare the performance of all proposed hierarchical feature selection methods combined with these different types of classifiers.

Recall that the proposed hierarchical feature selection methods are HIP, MR, HIP–MR and HRE–TAN. Among those four methods, only HIP–MR cannot eliminate all hierarchical redundancy, and it shows comparatively lower predictive accuracy than the HIP and MR methods. Hence, we only consider the comparison among the HIP, MR and HRE–TAN methods. HIP and MR methods follow the filter approach, so they can be used with the NB, TAN and GO–BAN classifiers, whereas HRE–TAN is an embedded method to be used with TAN. We assembled all experimental results for these methods reported earlier, i.e. GMean values for HIP and MR working with NB, TAN and GO–BAN classifiers, along with the GMean value for HRE–TAN, as shown in Table 6.8, where the boldface figures denote the highest GMean values for each dataset. More precisely, in Table 6.8, the GMean values reported for HIP+NB and MR+NB were taken from Tables

4.10 – 4.13 and Tables 4.14 – 4.17 respectively in Chapter 4; the GMean values for HIP+TAN, MR+TAN and HRE–TAN are taken from Tables 5.1 – 5.4 and Tables 5.5 – 5.8 in Chapter 5; and the GMean values for HIP+GO–BAN and MR+GO–BAN were taken from Table 6.1 to 6.4 in this current chapter.

We analyse the results reported in Table 6.8 from 3 different perspectives, as follows. First, we focus on identifying the hierarchical feature selection method which most often produced the best GMean values in general (working with different types of classifiers), rather than identifying the best combination of feature selection method and classifier. From this perspective, we consider a feature selection method as the winner in a dataset if that method obtained the highest GMean value in that dataset, regardless of which classifier was used together with the feature selection method. Overall, the HIP method obtained the highest GMean value in 22 out of the 28 datasets, and was the clear winner from this perspective. The second best hierarchical feature selection method, i.e. MR, obtained the highest GMean value in only 6 datasets; whilst HRE–TAN did not obtain any highest GMean value.

Second, we focus on identifying the type of Bayesian network classifier which most often produced the best GMean values in general (working with different types of hierarchical feature selection methods). From this perspective, we consider a type of classifier as the winner in a dataset if that type of classifier obtained the highest GMean value in that dataset, regardless of which hierarchical feature selection method was used together with that type of classifier. Overall, the most successful type of classifier was NB, which obtained the highest GMean value in 13 out of the 28 datasets. Among these 13 cases, 11 involve the use of the HIP feature selection method, whilst the other two cases involve the use of the MR method. The GO–BAN classifier was almost as successful as NB, obtaining the highest GMean value in 11 datasets – in 8 cases with HIP and in the other 3 cases with MR. TAN was the least successful classifier, obtaining the highest GMean value in only 5 out of the 28 datasets – in 4 cases with HIP and in one case with MR.

Third, we focus on identifying the combination of hierarchical feature selection method and type of classifier which most often produced the best GMean values. In order to compare the different methods from this perspective, Figure 6.7 shows the average ranks (across the 28 datasets included in Table 6.8) for each pair of

TABLE 6.8 GMean Values of All Proposed Hierarchical Feature Selection Methods Working with Different Classifiers

Feature Types	HIP + NB	HIP + TAN	HIP + GO–BAN	MR + NB	MR + TAN	MR + GO–BAN	HRE–TAN
Caenorhabditis elegans Datasets							
BP	**63.9**	59.4	63.2	62.2	63.4	62.2	56.2
MF	48.6	46.6	48.0	**49.6**	46.5	47.4	41.7
CC	**59.5**	52.8	59.2	55.3	53.0	54.6	44.5
BP + MF	63.8	62.7	64.4	67.9	66.2	**68.3**	58.2
BP + CC	64.9	64.0	65.3	63.9	**66.0**	63.5	57.6
MF + CC	**60.1**	55.2	60.0	57.0	56.9	57.5	50.8
BP + MF + CC	63.0	**65.5**	65.3	62.8	64.5	63.1	59.2
Drosophila melanogaster Datasets							
BP	57.2	**64.8**	63.3	55.5	62.0	59.7	51.5
MF	60.5	48.8	56.9	59.7	58.8	**61.7**	59.8
CC	61.2	62.8	60.5	**67.1**	57.0	60.6	46.6
BP + MF	64.9	62.0	**67.8**	59.5	61.7	59.9	52.4
BP + CC	**69.1**	63.3	67.7	67.5	57.8	64.2	52.4
MF + CC	**72.2**	68.0	69.6	69.5	61.9	68.0	58.7
BP + MF + CC	**72.1**	69.3	71.1	69.9	68.3	71.8	62.6
Mus musculus Datasets							
BP	71.3	56.0	**72.8**	63.6	60.6	62.4	63.9
MF	59.8	**67.9**	62.0	57.2	67.3	58.8	59.4
CC	61.5	60.3	**63.4**	58.1	57.2	58.9	59.7
BP + MF	**70.6**	58.2	69.8	64.2	66.6	59.8	58.8
BP + CC	69.8	63.2	**71.2**	62.4	64.4	64.2	61.2
MF + CC	68.3	63.9	70.0	68.4	67.9	**70.2**	56.3
BP + MF + CC	**73.5**	61.3	**73.5**	65.3	66.4	65.9	61.3
Saccharomyces cerevisiae Datasets							
BP	**70.4**	62.4	69.7	53.5	51.1	54.7	43.2
MF	20.4	**46.0**	43.0	0.0	0.0	0.0	0.0
CC	49.2	46.2	**49.4**	43.7	44.5	44.1	34.2
BP + MF	**75.3**	65.5	72.6	45.6	43.2	45.7	50.6
BP + CC	**74.6**	68.2	70.4	58.2	51.7	59.1	50.1
MF + CC	50.5	50.3	**57.8**	39.3	31.0	35.0	31.3
BP + MF + CC	72.5	69.8	**75.1**	50.9	57.3	53.8	47.3

feature selection method and type of classifier. HIP+NB and HIP+GO–BAN obtained the same best average rank of 2.2, which is successively better than the average ranks obtained by MR+GO–BAN, HIP+TAN, MR+NB, MR+TAN and HRE–TAN. We also conducted a statistical significance test on those algorithms using Friedman test and Holm's *post-hoc* method, with the results shown in Table 6.9. In this table, the first column represents the name of the algorithms (a combination of different feature selection methods and classifiers); the second column represents the average rank of GMean values for those algorithms; the third column represents the p-values of corresponding algorithms and the four column represents the adjusted significance level by adopting Holm's *post-hoc* method. Since both HIP+NB and HIP+GO–BAN obtain the same highest average ranking of GMean value, either HIP+NB or HIP+GO–BAN can be adopted as the control algorithm to be compared with other algorithms. Here we chose HIP+NB as the control algorithm, and found that HIP+NB and HIP+GO–BAN significantly outperform other compared algorithms. Therefore, it is obvious that the HIP method is overall the best performing hierarchical feature selection method in this thesis.

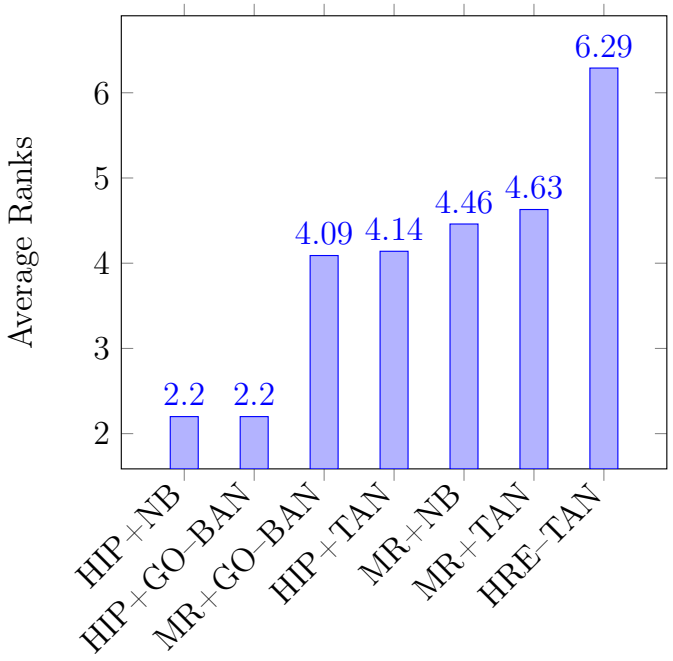


FIGURE 6.7 Average Ranks of Different Hierarchical Feature Selection Methods Working With Different Classifiers over 28 Datasets

TABLE 6.9 Statistical Significance Test Results of the Algorithms' GMean Values According to the Non-Parametric Friedman Test with the Holm *Post-Hoc* Test at the $\alpha = 0.05$ Significance Level

Algorithms	*Ave. Rank*	*P-value*	*Adjusted* α
HIP+NB (ctrl)	2.20	–	–
HIP+GO–BAN	2.20	1.0	0.0500
MR+GO–BAN	4.09	**1.06 E-03**	0.0250
HIP+TAN	4.14	**7.80 E-04**	0.0167
MR+NB	4.46	**9.08 E-05**	0.0125
MR+TAN	4.63	**2.57 E-05**	0.0100
HRE–TAN	6.29	**1.41 E-12**	0.0083

Chapter 7

Conclusions and Future Research Directions

The research described in this thesis is about hierarchical feature selection, which is a relatively new research subarea in machine learning/data mining. In this thesis, we proposed four hierarchical feature selection algorithms (three filter feature selection algorithms and one embedded feature selection algorithm), plus two network topology construction algorithms for Bayesian Network Augmented Naïve Bayes classifier based on the features selected by different feature selection algorithms (including conventional flat feature selection algorithms and the proposed hierarchical feature selection algorithms). All those algorithms have been empirically evaluated on datasets about the biology of ageing, and two of the best performing hierarchical feature selection algorithms have been applied to rank biological features in decreasing order of relevance for predicting ageing-related classes. Therefore, this research made contributions to both areas of machine learning/data mining and the biology of ageing.

Overall, the newly proposed hierarchical feature selection algorithms, which have been shown to be able to improve the predictive performance of Bayesian network classifiers, work with datasets where the features are hierarchically organised. In terms of those algorithms' application in this thesis, the objects being classified are genes, and the classes to be predicted indicate whether a gene has a "pro-longevity" or "anti-longevity" effect on an organism. Gene Ontology

(GO) terms are used as predictive features. These terms describe the functions of genes, and they are structured as a hierarchy (more precisely, a Directed Acyclic Graph (DAG)). Within the DAG, most of the GO terms follow a generalisation-specialisation relationship, which leads to redundancy between GO terms. Therefore, the proposed hierarchical feature selection methods aim at removing the redundancy within the hierarchy in order to improve the predictive performance of classifiers.

This chapter is organised as follows. In Section 7.1, the contributions of this thesis will be reviewed by summarising the newly proposed hierarchical feature selection algorithms and their use for ranking GO terms. In Section 7.2, future research directions will be proposed.

7.1 Contributions

The thesis made contributions to two areas, with the primary contributions being in the area of machine learning/data mining and secondary contribution to the biology of ageing. As mentioned at the beginning of this chapter, this thesis proposed four novel hierarchical feature selection algorithms and two network construction algorithms for Bayesian Network Augmented Naïve Bayes classifiers based on the features selected by either flat or hierarchical feature selection algorithms. Among those proposed algorithms, the two best performing hierarchical feature selection algorithms were used to rank GO terms (predictive features) in our ageing-related datasets.

7.1.1 Three Filter Hierarchical Feature Selection Algorithms

In Chapter 4, we proposed three hierarchical feature selection algorithms, namely select Hierarchical Information-Preserving (HIP) features, select Most Relevant (MR) features, and the hybrid select Hierarchical Information-Preserving and Most Relevant (HIP–MR) features. The HIP method eliminates all hierarchical redundancy by only selecting the features which retain all the hierarchical information on each individual path in the feature DAG; the MR method eliminates all hierarchical redundancy by only selecting the features which have the maximum relevance

value on each individual path; and the HIP–MR method merely alleviates (but does not completely eliminate) the hierarchical redundancy, since it removes the features whose relevance values are smaller than or equal to the relevance value of their corresponding ancestors or descendants. As shown in Table 7.1, which contains a summary on the merits and drawbacks of the three proposed algorithms, the HIP method eliminates hierarchical redundancy and selects a feature subset that retains all hierarchical information, whereas it ignores the relevance of individual features - since it does not consider any measure of association between a feature and the class attribute. The MR method eliminates hierarchical redundancy and selects features by considering both the hierarchical information and the features' relevance, but the selected features might not retain the complete hierarchical information. The HIP-MR method avoids the risk of losing hierarchical information and also considers the features' relevance, but it can merely alleviate (and not completely eliminate) hierarchical redundancy. In terms of the number of selected features, HIP selects the fewest, MR selects more, and HIP–MR selects the most. All those methods were evaluated by working with the Naïve Bayes classifier in Chapter 4.

TABLE 7.1 Summary on Proposed Hierarchical Feature Selection Methods

Hierarchical Feature Selection Algorithms	Merits	Drawbacks	Feature Selectivity
HIP	*Eliminate hierarchical redundancy; Retain all hierarchical information*	*Ignore relevance of features;*	*Select the smallest number of features*
MR	*Eliminate hierarchical redundancy; Select highly relevant features*	*Might lead to loss of hierarchical information*	*Select more features than HIP, less than HIP-MR*
HIP–MR	*Avoid loss of hierarchical information*	*Retain some redundancy among features*	*Select the largest number of features*

In details, we firstly proposed those three algorithms and evaluated them on datasets with ageing-related genes from four different model organisms, using as predictive features different combinations of three types of GO terms, namely biological process, molecular function and cellular component terms. Overall, those three proposed hierarchical feature selection methods improve the predictive performance of the Naïve Bayes classifier. In addition, for the purpose of further evaluating the predictive performance of the proposed algorithms, we conducted

comparisons between the two best performing algorithms (i.e. HIP and MR) and three conventional "flat" feature selection methods, namely Correlation-based feature selection (CFS), Entropy-based feature selection and Relevance-based feature selection, plus Naïve Bayes without feature selection. The experimental results reveal that HIP outperforms all other feature selection algorithms in terms of predictive accuracy, whereas MR's predictive accuracy does not show significant difference by comparison with other algorithms, except that MR significantly outperforms the Entropy-based feature selection.

We also further evaluated the performance of the HIP and MR methods from the perspective of robustness to imbalanced class distributions. The outcomes of this evaluation are that HIP is more robust than MR on dealing with the imbalanced class distribution issue, since the features selected by HIP tend to obtain relatively high values of both sensitivity and specificity; whereas MR tends to obtain much higher predictive accuracy when classifying the instances of the majority class than when classifying instances of the minority class, resulting in substantially imbalanced values of sensitivity and specificity.

Both these hierarchical feature selection algorithms, HIP and MR, were also further evaluated by using other types of Bayesian network classifiers, i.e. Tree Augmented Naïve Bayes Classifier and Bayesian Network Augmented Naïve Bayes Classifier in Chapter 5 and 6 respectively, as discussed below.

In addition, we also evaluated the computational running time for all proposed feature selection algorithms. In the experiments reported in this thesis the computational times were not large in general, mainly because, although the datasets had a large number of features, they had a relatively small number of instances. However, there are applications of feature selection and classification methods to protein function prediction problems where the number of instances is much larger (in addition to also having a large number of features). In such applications the issue of runtime of the proposed methods would be more relevant. In particular, Radivojac, et al. [100] discuss the results of a large-scale evaluation of computational protein function prediction methods, which was performed in the first international competition in this area, called Critical Assessment of protein Function Annotation (CAFA). As the number of organisms with known genome sequence keeps increasing, the number of corresponding proteins in databases like Uniprot keeps increasing too, and so the number of instances used in the datasets

of these international CAFA competitions (which are expected to continue to be held in the future) will also keep increasing.

7.1.2 An Embedded Hierarchical Feature Selection Algorithm for the Tree Augmented Naïve Bayes Classifier

In Chapter 5, we proposed one new embedded hierarchical feature selection method based on the Tree Augmented Naïve Bayes (TAN) Classifier, namely Hierarchical Redundancy Elimination-Tree Augmented Naïve Bayes (HRE–TAN). Briefly, this method removes the hierarchically redundant features during the processes of building the Maximum Spanning Tree, which is the main procedure used for building the Tree Augmented Naïve Bayes classifier.

We also conducted an empirical evaluation of this newly proposed algorithm on the datasets consisting of different combinations of the three types of GO terms mentioned earlier. Other compared algorithms evaluated in the experiments were the same methods adopted in Chapter 4, i.e. HIP, MR, CFS, Entropy-based and Relevance-based feature selection methods. We also compared those feature selection methods with the TAN classifier without any feature selection method. According to the comparison results, HIP again shows the best predictive performance and significantly outperforms all other feature selection methods except CFS when working with the TAN classifier. Analogously to HIP, MR significantly outperforms all compared feature selection methods except CFS and HIP.

In terms of the evaluation of robustness against imbalanced class distributions, when working with the TAN classifier, HIP again tends to obtain high values of both sensitivity and specificity simultaneously, whereas MR again tends to obtain much higher predictive accuracy when classifying the instances of the majority class than when classifying instances of the minority class.

7.1.3 Two Network Topology Construction Algorithms for Gene Ontology-Based Bayesian Network Augmented Naïve Bayes

In Chapter 6, we proposed two network topology construction algorithms for the Gene Ontology-based Bayesian Network Augmented Naïve Bayes (GO–BAN) classifier, based on the features selected by either flat or hierarchical feature selection methods. The first algorithm was proposed for GO–BAN working with a Flat Feature Selection method, and it is named FFS+GO–BAN; the second algorithm was proposed for GO–BAN working with a Hierarchical Feature Selection method, and it is named HFS+GO–BAN. Briefly, both these algorithms use the features which have already been selected by the corresponding feature selection methods, and they construct the dependencies (network edges) between those selected features according to the pre-defined dependencies on the GO DAG.

We conducted an empirical evaluation of the two proposed algorithms by using different feature selection methods, i.e. the hierarchical HIP and MR methods, the flat CFS method, and the GO–BAN classifier without any feature selection method, as a baseline. The results have shown that HIP+GO–BAN significantly outperforms all other GO–BAN methods.

In Chapter 6, we also conducted a further comparison involving all hierarchical feature selection methods proposed in Chapter 4, 5 and 6; combining them with different Bayesian network classifiers, i.e. NB, TAN and GO–BAN, from the perspective of their GMean values. The outcomes of this experimental comparison revealed that HIP+NB and HIP+GO–BAN significantly outperformed MR+GO–BAN, HIP+TAN, MR+NB, MR+TAN and HRE–TAN. It can be concluded that HIP is overall the best hierarchical feature selection method (among the methods evaluated in this thesis) for improving the predictive accuracy of two types of Bayesian network classifiers (i.e. NB and GO–BAN), when working with the data where the features are hierarchically organised.

7.1.4 Ageing-Related Dataset Creation and Ageing-Related GO Terms' Ranking

In terms of contributions to the biology of ageing, firstly, we created a set of ageing-related datasets referring to four model organisms, where genes are classified into pro-longevity or anti-longevity ones, using Gene Ontology (GO) terms as predictive features. This set of datasets is freely available for other researchers from [5]. Secondly, as another contribution of this thesis, we discovered some potentially interesting ageing-related patterns based on the proposed feature selection methods. In details, we created two rankings of GO terms in decreasing order of their usefulness for predicting the pro-longevity or anti-longevity class of a gene, for each model organism. The two rankings were mainly based on the frequency of selection of GO terms (features) by two hierarchical feature selection methods, i.e. HIP and MR. More precisely, for the GO terms selected by MR, we adopted the ranking criteria of selection frequency (as the main criterion) and statistical significance level (used as a tie-breaking criterion). For the GO terms selected by HIP working with the TAN classifier, we adopted as the main ranking criterion the selected frequency, and as a tie-breaking criterion the frequency of occurrence in the edges of the TAN classifier. Both ranking lists provide potentially insightful information about ageing research.

7.1.5 Computational Materials

The implementation of all proposed feature selection methods algorithms and classifiers was programmed in Java and Eclipse integrated development environment. Weka (Java-based source code) was used as a third-party source in the experiments using the correlation-based feature selection algorithm as a baseline method. Most experiments were run on a computer cluster generously provided by the School of Computing, University Kent. *I acknowledge the support of concurrence researchers at Kent for access to the 'CoSMoS' cluster, funded by EPSRC grants EP/E049419/1 and EP/E053505/1.*

The cluster was equipped with 12 nodes, each consists of two four-core Xeon

E5520 processors (16 hardware threads in total) and 12 GiB of RAM. The operating system used by the cluster was Ubuntu 12.04LTS. Very few experiments were run on an iMac equipped with one 2.9 GHz Intel Core i5 CPU, 2×4 GB 1600 MHz DDR3 memory, one Macintosh hard drive and OS X (version 10.8.2) operating system.

The datasets used in the experiments reported in Chapter 4 are available for downloading from the link: <http://www.cs.kent.ac.uk/people/staff/aaf/pub_papers.dir/IEEE_TCCB_Wan_Ageing_Datasets.zip>. The datasets used in the experiments reported in Chapters 5 and 6 are available from the author by request. The method used for generating all datasets is mentioned in Chapters 4 and 5, in pages 69, 88 and 119, respectively.

7.2 Future Research Directions

The future research directions suggested in this thesis can be categorised into six types. The first type includes research directions that are direct extensions of the work described in this thesis. In details, those proposed hierarchical feature selection methods can be further evaluated by combining them with other (non-Bayesian) types of lazy learning-based classifiers, e.g. Nearest Neighbour [108,123], lazy Decision Tree [40], etc. Actually, we have already performed some preliminary experiments evaluating HIP and MR with Nearest Neighbour classifiers, as reported in [123]; but these experiments involve only biological process GO terms. More experiments, with other types of GO terms (molecular function and cellular component terms), should also be performed.

Going beyond GO terms, the proposed hierarchical feature selection methods are generic enough to be applicable to any dataset with hierarchically organised features, as long as the hierarchical relationships represent generalisation-specialisation relationships. Hence, the proposed hierarchical methods should be further evaluated in other types of datasets too. For instance, these methods can be evaluated in text mining datasets, where instances represent documents, features typically represent the presence or absence of words in a document, and classes represent, for instance, the topic or subject of the document. Words also obey hierarchical, generalisation-specialisation relationships (as captured e.g. in

the WordNet system [31]), making text mining another natural application domain for the proposed hierarchical feature selection methods.

The second type of future research direction consists of proposing new embedded hierarchical feature selection methods based on lazy learning versions of other types of Bayesian network classifiers. For example, as mentioned in Chapter 2, the AODE classifier can be adapted to perform embedded hierarchical feature selection in order to alleviate hierarchical redundancy among features. More precisely, hierarchically redundant features can be removed for each individual One-Dependent Estimator (ODE) during the training phase of AODE. Then the classification phase of the conventional AODE classifier remains the same, i.e. the class predictions computed by the set of ODEs will be used for classifying a new testing instance.

The third type of future research direction consists of proposing a new lazy version of the CFS method [48], and then further extend lazy CFS to eliminate the hierarchical redundancy according to the pre-defined DAG in a way analogous to HIP and MR. In order to design lazy CFS, the calculation of the correlation coefficient between a pair of features, or between a feature and the class variable can be adapted for only considering the actual values of features on the current testing instance. Then, in order to incorporate hierarchical redundancy elimination into Lazy-CFS, during the stage of heuristic search for the most appropriate subset of features, the search space can be substantially reduced by removing hierarchically redundant features with respect to features in the current candidate feature subset.

The fourth type of future research directions consists of proposing other hierarchical feature selection methods that can be combined with eager learning-based classifiers, rather than only working with lazy learning-based classifiers. For example, one possible method would be firstly rank all features according to a certain eager learning-based measure of feature quality, and then remove features from the top to the bottom of the ranking, according to pre-defined hierarchical dependencies. Another possible method would be based on relaxing the definition of hierarchical redundancy, by measuring the degree of hierarchical redundancy between pairs of features. This could be measured by the degree of co-occurrence of pairs of features in the training set as a whole, from an eager learning perspective. Then a threshold could be chosen for deciding whether or not the degree of

co-occurrence is high enough to be considered a case of hierarchical redundancy. This new approach could be somehow integrated with the pre-defined hierarchical dependencies between features and then exploited by conducting hierarchical feature selection.

The fifth type of future research directions is an extension of the scenario when the classes or feature values are non-binary. The proposed hierarchical feature selection methods can be directly adopted for the multi-class classification task, where there are more than two class values. However, the performance of the proposed methods on this scenario still needs to be evaluated. In terms of the scenario of non-binary feature values, the proposed hierarchical feature selection methods cannot be directly adopted, since the definition of hierarchical redundancy in this thesis relies on binary feature values. Hence, new types of hierarchical feature selection methods should be developed, based on an extended definition of hierarchical feature redundancy for non-binary feature values.

The sixth type of future research directions is evaluating the usefulness of a feature hierarchy as a form of pre-defined expert knowledge, in the context of the classification task. As an example, in order to evaluate the usefulness of the Gene Ontology as a feature hierarchy, the proposed hierarchical feature selection methods could be applied to randomly generated variations of the feature hierarchy, e.g. randomly permuting the dependencies between GO terms.

In addition, in terms of future research direction on the application of hierarchical feature selection methods to the biology of ageing, it is suggested to create other datasets that contain other types of hierarchical features of genes or proteins, such as ageing-related pathway information by integrating data from the KEGG (Kyoto Encyclopedia of Genes and Genomes) database [67], Reactome [25], etc.

References

[1] A chain of amino acids. [online] Available at:<http://myhome.sunyocc.edu/~weiskirl/amino_acids_proteins.htm>, 2002. [Accessed on 11 August 2013].

[2] Fundamentals of protein structure, cs597a. [online] Available at:<http://www.cs.princeton.edu/courses/archive/fall07/cos597A/lectures/fundamentals.pdf>, 2007. [Accessed on 11 August 2013].

[3] Astbury in retrospect. [online] Available at:<http://www.leeds.ac.uk/heritage/hpsmuseum/astburyretro.htm>, 2011. [Accessed on 11 August 2013].

[4] Gene2go file. [online] Available at:<ftp://ftp.ncbi.nlm.nih.gov/gene/DATA/gene2go.gz>, 2012. [Accessed on 13 December 2012].

[5] Ageing-related genes datasets. [online] Available at:<http://www.cs.kent.ac.uk/people/staff/aaf/pub_papers.dir/IEEE_TCCB_Wan_Ageing_Datasets.zip>, 2014.

[6] D. W. Aha. *Lazy Learning*. Kluwer Academic Publishers, Norwell, MA, 1997.

[7] A. Al-Shahib, R. Breitling, and D. Gilbert. Feature selection and the class imbalance problem in predicting protein function from sequence. *Applied Bioinformatics*, 4(3):195–203, 2005.

[8] A. Al-Shahib, R. Breitling, and D. Gilbert. Franksum: new feature selection method for protein function prediction. *International Journal of Neural Systems*, 15(4):259–275, 2005.

[9] A. Alexa, J. Rahnenführer, and T. Lengauer. Improved scoring of functional groups from gene expression data by decorrelating GO graph structure. *Bioinformatics*, 22(13):1600–1607, Apr. 2006.

[10] C. F. Aliferis, A. Statnikov, I. Tsamardinos, S. Mani, and X. D. Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation. *The Journal of Machine Learning Research*, 11:171–234, 2010.

[11] D. G. Altman and J. M. Bland. Diagnostic tests. 1: Sensitivity and specificity. *BMJ: British Medical Journal*, 308(6943):1552, June 1994.

[12] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

[13] S. N. Austad. Retarded senescence in an insular population of virginia opossums (*Didelphis virginiana*). *Journal of Zoology*, 229(4):695–708, 1993.

[14] J. Bacardit, P. Widera, A. Márquez-Chamorro, F. Divina, J. S. Aguilar-Ruiz, and N. Krasnogor. Contact map prediction using a large-scale ensemble of rule sets and the fusion of multiple predicted structural features. *Bioinformatics*, 28(19):2441–2448, 2012.

[15] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 22(7):830–836, 2006.

[16] D. A. Benson, M. Cavanaugh, K. Clark, I. Karsch-Mizrachi, D. J. Lipman, J. O. J, and E. W. Sayers. Genbank. *Nucleic Acids Research*, 41:D36–D42, 2013.

[17] N. Bhardwaj, R. E. Langlois, G. Zhao, and H. Lu. Kernel-based machine learning protocol for predicting DNA-binding proteins. *Nucleic Acids Research*, 33(20):6486–6493, 2005.

[18] C. Bielza and P. Larrañaga. Discrete Bayesian network classifiers: A survey. *ACM Computing Surveys (CSUR)*, 47(1):5, July 2014.

[19] S. R. Bolsover, J. S. Hyams, S. Jones, E. A. Shephard, and H. A. White. *From Genes to Cells*. Wiley-Liss, New York, 1997.

[20] K. M. Borgwardt, C. S. Ong, S. Schönauer, S. V. N. Vishwanathan, A. J. Smola, and H. P. Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl 1):i47–i56, Mar. 2005.

[21] A. Brazma, H. Parkinson, T. Schlitt, and M. Shojatalab. A quick introduction to elements of biology-cells, molecules, genes, functional genomics, microarrays. [online] Available at:<http://www.ebi.ac.uk/microarray/biology-intro.html>, 2001. [Accessed on 11 November 2012].

[22] J. Campisi and F. D. A. di Fagagna. Cellular senescence: when bad things happen to good cells. *Nature Reviews Molecular Cell Biology*, 8(9):729–740, 2007.

[23] J. Cheng and R. Greiner. Comparing Bayesian network classifiers. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pages 101–108, Stockholm, Sweden, 1999.

[24] T. Craig, C. Smelick, R. Tacutu, D. Wuttke, S. H. Wood, H. Stanley, G. Janssens, E. Savitskaya, A. Moskalev, R. Arking, and J. P. de Magalhães. The digital ageing atlas: integrating the diversity of age-related changes into a unified resource. *Nucleic Acids Research*, 43:D873–D878, 2015.

[25] D. Croft, G. O'Kelly, G. Wu, R. Haw, M. Gillespie, L. Matthews, ..., and L. Stein. Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Research*, 39:D691–D697, 2011.

[26] J. P. de Magalhães. Programmatic features of aging originating in development: aging mechanisms beyond molecular damage? *The FASEB Journal*, 26(12):4821–4826, Dec. 2012.

[27] J. P. de Magalhães, A. Budovsky, G. Lehmann, J. Costa, Y. Li, V. Fraifeld, and G. M. Church. The human ageing genomic resources: online databases and tools for biogerontologists. *Aging Cell*, 8(1):65–72, Feb. 2009.

[28] J. Demsăr. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, Jan. 2006.

[29] J. Derrac, S. Garcia, D. Molina, and F. Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, 1(1):3–18, Mar. 2011.

[30] Y. Fang, X. Wang, E. K. Michaelis, and J. Fang. Classifying aging genes into DNA repair or non-DNA repair-related categories. In D. S. Huang, K. H. Jo, Y. Q. Zhou, and K. Han, editors, *Lecture Notes in Intelligent Computing Theories and Technology*, pages 20–29. Springer, Berlin Heidelberg, 2013.

[31] C. Fellbaum. *WordNet*. Blackwell Publishing Ltd, 1998.

[32] T. Finkel and N. J. Holbrook. Oxidants, oxidative stress and the biology of ageing. *Nature*, 408:239–247, Nov. 2000.

[33] T. Finkel, M. Serrano, and M. A. Blasco. The common biology of cancer and ageing. *Nature*, 448(7155):767–774, 2007.

[34] A. A. Freitas and J. P. de Magalhães. A review and appraisal of the DNA damage theory of ageing. *Mutation Research*, 728(1-2):12–22, July/Oct. 2011.

[35] A. A. Freitas, O. Vasieva, and J. P. de Magalhães. A data mining approach for classifying DNA repair genes into ageing-related or non-ageing-related. *BMC Genomics*, 12(27):1–11, Jan. 2011.

[36] A. A. Freitas, D. C. Wieser, and R. Apweiler. On the importance of comprehensible classification models for protein function prediction. *IEEE/ACM Transections on Computational Biology and Bioinformatics*, 7(1):172–182, 2010.

[37] A. A. Freitas. *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer, Berlin/Heidelberg, 2002.

[38] A. A. Freitas. Comprehensible classification models - a position paper. *ACM SIGKDD Explorations*, 15(1):1–10, June 2013.

[39] I. Friedberg. Automated protein function prediction-the genomic challenge. *Briefings in Bioinformatics*, 7(3):225–242, Jan. 2006.

[40] J. H. Friedman, R. Kohavi, and Y. Yun. Lazy decision trees. In *Proceedings of Thirteenth National Conference on Artificial Intelligence*, pages 717–724, Portland, USA, Aug. 1996.

[41] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29(2-3):131–163, Nov. 1997.

[42] S. Fu and M. C. Desmarais. Markov blanket based feature selection: a review of past decade. In *Proceedings of the World Congress on Engineering 2010*, volume 1, pages 321–328, London, UK, 2010.

[43] L. A. Gavrilov and N. S. Gavrilova. Evolutionary theories of aging and longevity. *The Scientific World Journal*, 2:339–356, 2002.

[44] Z. Ghahramani. Probabilistic machine learning and artificial intelligence. *Nature*, 521(7553):452–459, May 2015.

[45] E. Glaab, J. Bacardit, J. M. Garibaldi, and N. Krasnogor. Using rule-based machine learning for candidate disease gene prioritization and sample classification of cancer gene expression data. *PLoS One*, 7:e39932, 2012.

[46] L. Guarente and C. Kenyon. Genetic pathways that regulate ageing in model organisms. *Nature*, 408(6809):255–262, Nov. 2000.

[47] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, Mar. 2003.

[48] M. A. Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999. PhD Thesis.

[49] M. A. Hall and L. A. Smith. Feature subset selection: a correlation based filter approach. In *Proceedings of 1997 International Conference on Neural Information Processing and Intelligent Information Systems*, pages 855–858, 1997.

[50] J. Han, M. Kamber, and J. Pei. *Data mining: concepts and techniques: concepts and techniques*. Elsevier, San Francisco, CA, 2011.

[51] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[52] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Springer-Verlag, 2009.

[53] L. K. Heilbronn and E. Ravussin. Calorie restriction and aging: review of the literature and implications for studies in humans. *The American Journal of Clinical Nutrition*, 78(3):361–369, Sept. 2003.

[54] T. Huang, J. Zhang, Z. P. Xu, L. L. Hu, L. Chen, J. L. Shao, L. Zhang, X. Y. Kong, Y. D. Cai, and K. C. Chou. Deciphering the effects of gene deletion on yeast longevity using network and machine learning approaches. *Biochimie*, 94(4):1017–1025, Apr. 2012.

[55] N. Hurwitz, M. Pellegrini-Calace, and D. T. Jones. Towards genome-scale structure prediction for transmembrane proteins. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 361(1467):465–475, 2006.

[56] Imgur. Regression example. [online] Available at:<http://i.stack.imgur.com/t491d.png>. [Accessed on 8 June 2015].

[57] N. H. G. R. Institute. Biological pathways. [online] Available at:<http://www.genome.gov/27530687>, 2012. [Accessed on 19 June 2013].

[58] N. Japkowicz and M. Shah. *Evaluating learning algorithms: a classification perspective*. Cambridge University Press, New York, USA, 2011.

[59] R. Jenatton, J. Y. Audibert, and F. Bach. Structured variable selection with sparity-inducing norms. *Journal of Machine Learning Research*, 12:2777–2824, 2011.

[60] L. J. Jensen, M. Kuhn, M. Stark, S. Chaffron, C. Creevey, J. Muller, T. Doerks, P. Julien, A. Roth, M. Simonovic, P. Bork, and C. von Mering. String 8-a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Research*, 37(suppl 1):D412–D416, 2009.

[61] X. Jin and J. Han. *Encyclopedia of Machine Learning*. Springer US, 2010.

[62] D. T. Jones. *Protein Structure Prediction*, chapter A practical guide to protein structure prediction. Humana Press, Totowa, 2000.

[63] D. T. Jones, D. W. A. Buchan, D. Cozzetto, and M. Pontil. Psicov: Precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

[64] S. Jungjit and A. A. Freitas. A new genetic algorithm for multi-label correlation-based feature selection. In *Proceedings of the Twenty-Third European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN-2015)*, pages 285–290, Bruges, Belgium, 2015.

[65] S. Jungjit and A. A. Freitas. A lexicographic multi-objective genetic algorithm for multi-label correlation-based feature selection. In *Proceedings of the Companion Publication of Workshop on Evolutionary Rule-based Machine Learning at the Genetic and Evolutionary Computation Conference (GECCO 2015)*, pages 989–996, Madrid, Spain, 2015.

[66] R. Kaletsky and C. T. Murphy. The role of insulin/igf-like signaling in *C. elegans* longevity and aging. *Disease Models and Mechanisms*, 3(7-8):415–419, 2010.

[67] M. Kanehisa and S. Goto. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 2000.

[68] C. J. Kenyon. The genetics of ageing. *Nature*, 464(7288):504–512, Mar. 2010.

[69] E. J. Keogh and M. J. Pazzani. Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches. In *Proceedings of the Seventh International Workshop on Artificial Intelligence and Statistics*, pages 225–230, Florida, USA, Jan. 1999.

[70] T. B. L. Kirkwood. Understanding the odd science of aging. *Cell*, 120(4):437–447, Feb. 2005.

[71] T. B. L. Kirkwood and S. N. Austad. Why do we age? *Nature*, 408(6809):233–238, Nov. 2000.

[72] R. Kohavi. Scaling up the accuracy of naive-Bayes classifiers: A decision-tree hybrid. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 96)*, pages 202–207, Portland, USA, Aug. 1996.

[73] I. Kononenko. Semi-naive Bayesian classifier. In *Proceedings of Machine Learning-European Working Session on Learning*, pages 206–219, Porto, Portugal, Mar. 1991.

[74] T. Kosciolek and D. T. Jones. *De Novo* structure prediction of globular proteins aided by sequence variation-derived contacts. *PLoS One*, 9(3):e92197, Mar. 2014.

[75] V. G. Krishnan and D. R. Westhead. A comparative study of machine-learning methods to predict the effects of single nucleotide polymorphisms on protein function. *Bioinformatics*, 19(17):2199–2209, 2003.

[76] P. Langley and S. Sage. Induction of selective Bayesian classifiers. In *Proceedings of the Tenth International Conference on Uncertainty in Artificial Intelligence*, pages 399–406, Seattle, USA, 1994.

[77] R. A. Laskowski, J. D. Watson, and J. M. Thornton. Protein function prediction using local 3D templates. *Journal of Molecular Biology*, 351(3):614–626, Aug. 2005.

[78] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. Prentice-hall, Englewood Cliffs, 1974.

[79] I. Lee, B. Lehner, C. Crombie, W. Wong, A. G. Fraser, and E. M. Marcotte. A single gene network accurately predicts phenotypic effects of gene perturbation in *Caenorhabditis elegans*. *Nature Genetics*, 40:181–188, Jan. 2008.

[80] F. Leijoto, Larissa, T. A. D. O. Rodrigues, L. E. Zaratey, and C. N. Nobre. A genetic algorithm for the selection of features used in the prediction of

protein function. In *Proceedings of 2014 IEEE International Conference on Bioinformatics and Bioengineering (BIBE-2014)*, pages 168–174, Boca Raton, USA, Nov. 2014.

[81] B. Q. Li, L. L. Hu, L. Chen, K. Y. Feng, Y. D. Cai, and K. C. Chou. Prediction of protein domain with mRMR feature selection and analysis. *PLoS One*, 7(6):e39308, 2012.

[82] Y. H. Li, M. Q. Dong, and Z. Guo. Systematic analysis and prediction of longevity genes in caenorhabditis elegans. *Mechanisms of Ageing and Development*, 131(11-12):700–709, Nov./Dec. 2010.

[83] N. J. Linford, T. H. Kuo, T. P. Chan, and S. D. Pletcher. Sensory perception and aging in model systems: From the outside in. *Cell and Developmental Biology*, 27:759–785, Nov. 2011.

[84] H. Liu and H. Motoda. *Feature extraction, construction and selection: A data mining perspective*. Springer US, 1998.

[85] C. López-Otín, M. A. Blasco, L. Partridge, M. Serrano, and G. Kroemer. The hallmarks of aging. *Cell*, 153(6):1194–1217, June 2013.

[86] K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic press, 1979.

[87] A. F. T. Martins, N. A. Smith, P. M. Q. Aguiar, and M. A. T. Figueiredo. Structured sparsity in structured prediction. In *Proceeding of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*, pages 1500–1511, Edinburgh, UK, July 2011.

[88] E. J. Masoro. Overview of caloric restriction and ageing. *Mechanisms of Ageing and Development*, 126(9):913–922, Sept. 2005.

[89] Mathworks. Classification example. [online] Available at:<http://www.mathworks.com/matlabcentral/fileexchange/screenshots/9190/original.jpg>. [Accessed on 8 June 2015].

[90] C. M. McCay, M. F. Crowell, and L. A. Maynard. The effect of retarded growth upon the length of life span and upon the ultimate body size. *Journal of Nutrition*, 10(1):63–79, 1935.

[91] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3:235–244, 4.

[92] M. Minsky. Steps toward artificial intelligence. In *Proceedings of the IRE*, pages 8–30, Jan. 1961.

[93] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, Cambridge, MA, 2012.

[94] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1226–1238, Aug. 2005.

[95] M. A. Peot. Geometric implications of the naive Bayes assumption. In *Proceedings of the Twelfth International Conference on Uncertainty in Artificial Intelligence*, pages 414–419, Portland, USA, aug 1996.

[96] R. B. Pereira, A. Plastino, B. Zadrozny, L. H. de C. Merschmann, and A. A. Freitas. Lazy attribute selection: Choosing attributes at classification time. *Intelligent Data Analysis*, 15(5):715–732, Aug. 2011.

[97] M. Plank, D. Wuttke, S. van Dam, S. A. Clarke, and J. P. de Magalhães. A meta-analysis of caloric restriction gene expression profiles to infer common signatures and regulatory mechanisms. *Molecular Biosystems*, 8(4):1339–1349, Feb. 2012.

[98] T. S. K. Prasad and et al. Human protein reference database - 2009 update. *Nucleic Acids Research*, 37(suppl 1):D767–D772, Nov. 2009.

[99] Pypr. Cluster example. [online] Available at:<http://pypr.sourceforge.net/kmeans.html>. [Accessed on 8 June 2015].

[100] P. Radivojac, W. T. Clark, T. R. Oron, A. M. Schnoes, T. Wittkop, A. Sokolov, ..., and C. Schaefer. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, 2013.

[101] S. Raha and B. H. Robinson. Mitochondria, oxygen free radicals, disease and ageing. *Trends in Biochemical Sciences*, 25(10):502–508, 2000.

[102] R. J. Reece. *Analysis of Genes and Genomes.* John Wiley & Sons Ltd, Chichester, 2004.

[103] E. Rich and K. Knight. *Artificial intelligence.* McGraw-Hill Publishing Co., 1991.

[104] R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 3(1):88, Mar. 2007.

[105] J. Shen, L. Li, and W. K. Wong. Markov blanket feature selection for support vector machines. In *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence 2008*, volume 8, pages 696–701, Chicago, USA, 2008.

[106] R. S. Sohal, H. H. Ku, S. Agarwal, M. J. Forster, and H. Lal. Oxidative damage, mitochondrial oxidant generation and antioxidant defenses during aging and in response to food restriction in the mouse. *Mechanisms of Ageing and Development*, 74(1-2):121–133, May 1994.

[107] R. S. Sohal and R. Weindruch. Oxidative stress, caloric restriction, and aging. *Science*, 273(5271):59–63, July 1996.

[108] C. Stanfill and D. Waltz. Toward memory-based reasoning. *Communications of the ACM*, 29(12):1213–1228, Dec. 1986.

[109] C. Stark, B. J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. Biogrid: a general repository for interaction datasets. *Nucleic Acids Research*, 34:D535–D539, 2006.

[110] T. Strutz. *Data Fitting and Uncertainty (A practical introduction to weighted least squares and beyond).* Vieweg+Teubner, Wiesbaden, 2010.

[111] R. Tacutu, T. Craig, A. Budovsky, D. Wuttke, G. Lehmann, D. Taranukha, J. Costa, V. E. Fraifeld, and J. P. de Magalhães. Human ageing genomic resources: Integrated databases and tools for the biology and genetics of ageing. *Nucleic Acids Research*, 41(D1):D1027–D1033, Jan. 2013.

[112] The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000.

[113] R. Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society, Series B (Methodological)*, 58(1):267–288, 1996.

[114] I. Tsamardinos, C. F. Aliferis, A. R. Statnikov, and E. Statnikov. Algorithms for large scale Markov blanket discovery. In *Proceedings of FLAIRS Conference 2003*, volume 2, pages 376–380, St. Augustine, Florida, USA, May 2003.

[115] P. C. Turner, A. G. McLennan, A. D. Bates, and M. R. H. White. *Molecular Biology*. BIOS Scientific Publishers Ltd, Oxford, 2 edition, 2000.

[116] S. D. Tyner, S. Venkatachalam, J. Choi, S. Jones, N. Ghebranious, H. Igelmann, X. Lu, G. Soron, B. Cooper, C. Brayton, S. H. Park, T. Thompson, G. Karsenty, A. Bradley, and L. A. Donehower. P53 mutant mice that display early ageing-associated phenotypes. *Nature*, 415(6867):45–53, Jan. 2002.

[117] University of Michigan. Gene expression. [online] Available at:<http://biosocialmethods.isr.umich.edu/epigenetics-tutorial/epigenetics-tutorial-gene-expression-from-dna-to-protein/>. [Accessed on 1 July 2015].

[118] University of Warwick. Gene. [online] Available at:<http://www2.warwick.ac.uk/fac/sci/maths/research/events/2008_2009/workshops/isscngc/>. [Accessed on 1 July 2015].

[119] U.S. National Library of Medicine. DNA double helix. [online] Available at:<http://ghr.nlm.nih.gov/handbook/basics/dna>. [Accessed on 11 August 2013].

[120] J. Vijg and J. Campisi. Puzzles, promises and a cure for ageing. *Nature*, 454(7208):1065–1071, Aug. 2008.

[121] G. Walker, K. Houthoofd, J. R. Vanfleteren, and D. Gems. Dietary restriction in *C. elegans*: from rate-of-living effects to nutrient sensing pathways. *Mechanisms of Ageing and Development*, 126(9):929–937, Sept. 2005.

[122] C. Wan and A. A. Freitas. Prediction of the pro-longevity or anti-longevity effect of *Caenorhabditis Elegans* genes based on Bayesian classification methods. In *Proceedings of IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2013)*, pages 373–380, Shanghai, China, Dec. 2013.

[123] C. Wan, A. A. Freitas, and J. P. de Magalhães. Predicting the pro-longevity or anti-longevity effect of model organism genes with new hierarchical feature selection methods. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(2):262–275, Mar. 2015.

[124] C. Wan and A. A. Freitas. Two methods for constructing a gene ontology-based feature selection network for a Bayesian network classifier and applications to datasets of aging-related genes. In *Proceedings of the Sixth ACM Conference on Bioinformatics, Computational Biology and Health Informatics (ACM-BCB 2015)*, pages 27–36, Atlanta, USA, Sept. 2015.

[125] C. Wan and A. A. Freitas. Gene ontology hierarchy-based feature selection. In *Features and Structures 2014 (FEAST 2014) Workshop Attached to 22nd International Conference on Pattern Recognition (ICPR 2014)*, Aug. 2014.

[126] Z. Wang and G. I. Webb. Comparison of lazy Bayesian rule, and tree-augmented Bayesian learning. In *Proceedings of IEEE International Conference on Data Mining (ICDM 2002)*, pages 490–497, Maebashi, Japan, Dec. 2002.

[127] G. I. Webb, J. R. Boughton, and Z. Wang. Not so naive Bayes: aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.

[128] D. Wieser, I. Papatheodorou, M. Ziehm, and J. M. Thornton. Computational biology for ageing. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 366(1561):51–63, 2011.

[129] I. H. Witten, E. Frank, and M. A. Hall. *Data Mining: practical machine learning tools and techniques*. Morgan Kaufmann, Burlington, MA, 2011.

[130] J. G. Wood, B. Rogina, S. Lavu, K. Howitz, S. L. Helfand, M. Tatar, and D. Sinclair. Sirtuin activators mimic caloric restriction and delay ageing in metazoans. *Nature*, 430:686–689, July 2004.

[131] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning (ICML 1997)*, pages 412–420, Nashville, USA, 1997.

[132] S. Yaramakala and D. Margaritis. Speculative Markov blanket discovery for optimal feature selection. In *Proceedings of the Fifth IEEE International Conference on Data mining (ICDM)*, Houston, USA, Nov. 2005.

[133] J. Ye, G. Coulouris, I. Zaretskaya, I. Cutcutache, S. Rozen, and T. L. Madden. Primer-blast: a tool to design target-specific primers for polymerase chain reaction. *BMC Bioinformatics*, 13(1):134, 2012.

[134] J. Ye and J. Liu. Sparse methods for biomedical data. *ACM SIGKDD Explorations Newsletter*, 14(1):4–15, June 2012.

[135] J. Ye, N. Ma, T. L. Madden, and J. M. Ostell. Igblast: an immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Research*, pages W34–40, May 2013.

[136] M. Yousef, S. Jung, A. V. Kossenkov, L. C. Showe, and M. K. Showe. Naive Bayes for microrna target predictions–machine learning for microrna targets. *Bioinformatics*, 23(22):2987–2992, 2007.

[137] L. Yu and H. Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *Proceedings of the Twentieth International Conference on Machine Learning (ICML 2003)*, Washington DC, USA, Aug. 2003.

[138] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *The Journal of Machine Learning Research*, 5:1205–1224, 2004.

[139] Y. Zeng, J. Luo, and S. Lin. Classification using Markov blanket for feature selection. In *Proceedings of IEEE International Conference on Granular Computing, 2009, GRC '09*, pages 743–747, Nanchang, China, Aug. 2009.

[140] P. Zhao, G. Rocha, and B. Yu. The composite absolute penalties family for grouped and hierarchical variable selection. *The Annual of Statistics*, 37(6):3468–3497, 2009.

[141] F. Zheng and G. I. Webb. A comparative study of semi-naive Bayes methods in classification learning. In *Proceedings of the Fourth Australasian Data Mining Conference (AusDM05)*, pages 141–155, Sydney, Australia, Dec. 2005.

[142] F. Zheng and G. I. Webb. Efficient lazy elimination for averaged one-dependence estimators. In *Proceedings of the Twenty-Third International Conference on Machine Learning (ICML 2006)*, pages 1113–1120, Pittsburgh, USA, June 2006.

[143] Z. Zheng and G. I. Webb. Lazy learning of Bayesian rules. *Machine Learning*, 41(1):53–84, 2000.