New Multi-Label Correlation-Based Feature Selection Methods for Multi-Label Classification and Application in Bioinformatics

A Thesis submitted to
the University of Kent at Canterbury
in the subject of Computer Science
for the degree of
Doctor of Philosophy

By
Suwimol Jungjit
March 2016

Abstract

The very large dimensionality of real world datasets is a challenging problem for classification algorithms, since often many features are redundant or irrelevant for classification. In addition, a very large number of features leads to a high computational time for classification algorithms. Feature selection methods are used to deal with the large dimensionality of data by selecting a relevant feature subset according to an evaluation criterion. The vast majority of research on feature selection involves conventional single-label classification problems, where each instance is assigned a single class label; but there has been growing research on more complex multi-label classification problems, where each instance can be assigned multiple class labels.

This thesis proposes three types of new Multi-Label Correlation-based Feature Selection (ML-CFS) methods, namely: (a) methods based on hill-climbing search, (b) methods that exploit biological knowledge (still using hill-climbing search), and (c) methods based on genetic algorithms as the search method.

Firstly, we proposed three versions of ML-CFS methods based on hill climbing search. In essence, these ML-CFS versions extend the original CFS method by extending the merit function (which evaluates candidate feature subsets) to the multi-label classification scenario, as well as modifying the merit function in other ways. A conventional search strategy, hill-climbing, was used to explore the space of candidate solutions (candidate feature subsets) for those three versions of ML-

CFS. These ML-CFS versions are described in detail in Chapter 4.

Secondly, in order to try to improve the performance of ML-CFS in cancer-related microarray gene expression datasets, we proposed three versions of the ML-CFS method that exploit biological knowledge. These ML-CFS versions are also based on hill-climbing search, but the merit function was modified in a way that favours the selection of genes (features) involved in pre-defined cancer-related pathways, as discussed in detail in Chapter 5.

Lastly, we proposed two more sophisticated versions of ML-CFS based on Genetic Algorithms (rather than hill-climbing) as the search method. The first version of GA-based ML-CFS is based on a conventional single-objective GA, where there is only one objective to be optimized; while the second version of GA-based ML-CFS performs lexicographic multi-objective optimization, where there are two objectives to be optimized, as discussed in detail in Chapter 6.

In this thesis, all proposed ML-CFS methods for multi-label classification problems were evaluated by measuring the predictive accuracies obtained by two well-known multi-label classification algorithms when using the selected features namely: the Multi-Label K-Nearest neighbours (ML-kNN) algorithm and the Multi-Label Back Propagation Multi-Label Learning Neural Network (BPMLL) algorithm.

In general, the results obtained by the best version of the proposed ML-CFS methods, namely a GA-based ML-CFS method, were competitive with the results of other multi-label feature selection methods and baseline approaches. More precisely, one of our GA-based methods achieved the second best predictive accuracy out of all methods being compared (both with ML-kNN and BPMLL used as classifiers), but there was no statistically significant difference between that GA-based ML-CFS and the best method in terms of predictive accuracy. In addition, in the

experiment with ML-kNN (the most accurate) method selects about twice as many features as our GA-based ML-CFS; whilst in the experiments with BPMLL the most accurate method was a baseline method that does not perform any feature selection, and runs the classifier once (with all original features) for each of the many class labels, which is a very computationally expensive baseline approach.

In summary, one of the proposed GA-based ML-CFS methods managed to achieve substantial data reduction, (selecting a smaller subset of relevant features) without a significant decrease in predictive accuracy with respect to the most accurate method.

Acknowledgements

Firstly, I would like to thank my supervisor Prof. Alex A. Freitas for the continuous support of my Ph.D study and related research, for his kindness, patience and motivation. His guidance helped me in all the time of this research and the writing of this thesis. I could not have imagined having a better supervisor for my Ph.D study.

Besides my advisor, I would like to thank the rest of my thesis supervisory panel: Prof. Howard Bowman and Dr. Peter Rodgers, for their comments, discussion and the comprehensive questions which motivated me to expand my research from various perspectives. Also, I would like to thank Prof. Martin Michaelis (School of Bioscience at University of Kent) for providing microarray gene expression datasets used in this research.

I also would like to thank the Thai Royal Government which provided me with a full Ph.D studentship and an opportunity to study abroad. Without their support, I would not be able to undertake my study and do this research.

Last but not the least, I would like to thank my family: my parents, my sister, my nephew and my lovely friends for supporting me spiritually throughout writing this thesis and my life in general.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

1.1.1 Feature Selection in Data Pre-processing

Feature selection is a type of data pre-processing method (a part of the broader process of Knowledge Discovery [109]) which aims to select a relevant feature subset according to an evaluation criterion [76]. In the real world, the amount of stored data grows significantly and fast in many application domains. For example, microarray gene expression data analysis and document classification (a type of text mining) are applications where datasets usually have thousands of features. As a result, the very large dimensionality of the data is a crucial challenge for classification algorithms. A very large number of features leads to a high computational time for the classification algorithm and often most features are irrelevant or noisy, potentially leading to an overfitting of the classification model to the data [39, 115].

Feature selection methods directly address the large dimensionality of the data in the data pre-processing phase. A variety of feature selection methods has been proposed in the literature. Broadly speaking, feature selection for classification can be done using two approaches [19, 20, 25, 39, 44, 67, 76, 77, 79, 90, 97]: an embedded approach (where the feature selection process is performed during the

run of a classification algorithm) or a data preprocessing approach, where a feature subset is selected and then given to the classification algorithm. This research focuses on the data preprocessing approach, which is more generic than the embedded approach.

1.1.2 Multi-Label Classification

Classification is a data mining task where the system is given a dataset of instances (records, objects) – each one described by a set of feature values and belonging to a class – and then the system has to extract, from the dataset, a classification model that predicts the class value (label) for an unseen instance, given the values of the features describing that instance [116].

Multi-label classification is different from traditional single-label classification because in multi-label classification each instance can be associated with a set of class labels [21, 24, 92], while in traditional single-label classification each instance is associated with only one class label.

The vast majority of research projects in classification involve single-label classification. However, there is a growing research trend in tackling the more difficult problem of multi-label classification [15, 24, 26, 103, 105, 108, 111, 113]. This is motivated by a number of real-world classification problems that are naturally described as multi-label problems. For example, an article about social media can be classified to both information technology and social activity class labels. A document can be classified to the class labels education and linguistic at the same time. A gene can be associated with many biological functions in an organism, and an image can be annotated with sea, forest and mountain class labels.

1.2 The Goals and the Focus of This Research

This research focuses on feature selection, with the two related goals of proposing new multi-label feature selection algorithms for multi-label classification problems and evaluating the proposed algorithms' predictive performance in a set of multi-label datasets. The major research question addressed by this thesis is whether the proposed multi-label feature selection methods can select the most relevant and non-redundant features and improve the predictive accuracy when compared with other multi-label feature selection methods in the literature.

Most of the multi-label feature selection methods proposed in this thesis are generic in the sense that they can be applied to multi-label classification datasets from any application domain. However, Chapter 5 of this thesis proposes feature selection methods specially designed for exploiting biological knowledge about cancer-related pathways, in microarray gene expression datasets having more than 20,000 features, as will be described later.

1.3 Original Contributions

This thesis propose three types of new Multi-Label Correlation-based Feature Selection (ML-CFS) methods, namely: (a) methods based on hill-climbing search, (b) methods that exploit biological knowledge (still using hill-climbing search), and (c) methods based on genetic algorithms as the search method. The summary of the original contributions of this thesis is presented in Figure 1.1.

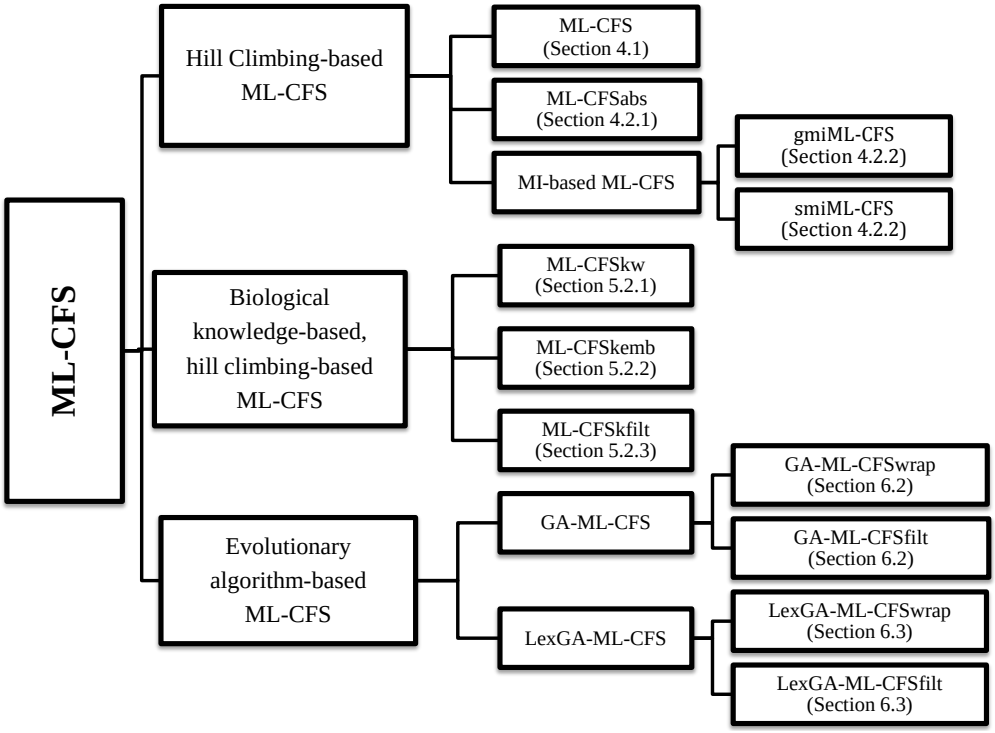


Figure 1.1: Summary of Original Contributions: ML-CFS methods

1.3.1 Multi-Label Correlation-Based Feature Selection (ML-CFS) Methods Based on Hill Climbing Search

We propose three versions of a new ML-CFS method that is an extension of the single-label CFS method proposed by Hall [44] to the multi-label classification problem. In essence, these ML-CFS versions extend the original CFS method by extending the merit function used by the method to evaluate candidate solutions, as discussed in detail in Chapter 4. These versions have in common the fact that they use the same conventional search strategy to explore the space of candidate solutions (candidate feature subsets), namely a well-known hill-climbing search strategy.

1.3.2 Multi-Label Correlation-Based Feature Selection (ML-CFS) Methods that Exploit Biological Knowledge

We propose three versions of the ML-CFS method that exploit biological knowledge, more precisely, the knowledge that some genes are involved in cancer-related pathway. The motivation for this is to try to improve the performance of ML-CFS in cancer-related microarray gene expression datasets, where features represent genes, so that feature selection corresponds to selecting relevant genes to predict a cancer-related class label. These ML-CFS versions are also based on hill-climbing search (like the versions mentioned in the previous subsection), but they modify ML-CFS' evaluation function or the original set of features in a way that favours the selection of genes (features) involved in pre-defined cancer-related pathways, as discussed in detail in Chapter 5.

1.3.3 Multi-Label Correlation-Based Feature Selection (ML-CFS) Methods Based on Evolutionary Algorithms

We propose two versions of a Genetic Algorithm (GA)-based ML-CFS method, denoted GA-ML-CFS. These versions replace the simple hill-climbing search method used by the previous ML-CFS versions by a more sophisticated GA. The first version of GA-ML-CFS is based on a conventional single-objective GA, where there is only one objective to be optimized, namely ML-CFS' evaluation function. The second version of GA-ML-CFS is based on a somewhat more sophisticated type of GA that performs lexicographic multi-objective optimization, where there are two objectives to be optimized – namely ML-CFS' evaluation function and the number of selected features – and the objectives are optimized in decreasing order of priority (called a lexicographic approach), as discussed in detail in Chapter 6.

1.4 The Structure of the Thesis

This Section outlines the structure of the remaining Chapters of this thesis. In essence, Chapters 2 and 3 describe the background on classification, feature selection and multi-label classification. Chapters 4 through 6 describe all versions of the Multi-Label Correlation-based Feature Selection (ML-CFS) methods proposed in this thesis. The summaries of Chapters 2 through 7 are as follows:

Chapter 2 – Background on Conventional Single-Label Classification and Bioinformatics: This chapter will contain background about knowledge discovery and data mining, focusing on conventional single-label feature selection for the classification task. In particular, it will review both hill-climbing-based and evolutionary algorithm-based methods for feature selection. It will also describe the single-label correlation-based feature selection (CFS) method proposed by Hall [44], which was the inspiration for the multi-label feature selection methods proposed in this thesis. This chapter will also briefly describe general background on bioinformatics and molecular biology, especially on microarray gene expression data and other bioinformatics topics related with our research (e.g. KEGG Pathway). This background is relevant for a better understanding of Chapter 5, which focuses on feature selection for gene expression data.

Chapter 3 – Background on Multi-Label Classification and Multi-Label Feature Selection: This chapter will present a survey of the multi-label classification area. It will include multi-label problem transformation methods, which transform a multi-label classification problem into one or more conventional (single-label) classification problems. In addition, two well-known multi-label classification algorithms which are used in this thesis will be described: the Multi-Label K-Nearest neighbours (ML-kNN) algorithm and the Multi-Label Back Propagation Multi-Label Learning Neural Network (BPMLL) algorithm. Also, multi-label classification evaluation measures, and multi-label feature selection methods, will

be reviewed.

Chapter 4 – The Proposed Multi-Label Correlation-Based Feature Selection Method Based on Hill Climbing Search: This Chapter will propose new correlation-based feature selection methods, which are extensions from the single-label CFS proposed by Hall [44], for the multi-label scenario. All ML-CFS versions described in this Chapter are based on hill-climbing search. In general, we describe the first version of ML-CFS and also other two types of extended ML-CFS versions: (1) ML-CFS with the absolute value of correlation coefficient and (2) ML-CFS using mutual information for class label weighting. This Chapter will also present all details of our experiments; such as dataset descriptions, the results from extensive experiments, comparisons with the results of other multi-label feature selection methods, and the corresponding discussion of those results.

Chapter 5 – The Proposed Multi-Label Correlation-Based Feature Selection Methods that Exploit Biological Knowledge: This Chapter will describe the extensions of ML-CFS specific to microarray gene expression datasets, which use background biological knowledge to help to guide the search for good feature subsets. This Chapter proposes three extensions of the ML-CFS method, involving three different approaches to exploit knowledge about cancer-related genes to select the most relevant features (genes) in microarray gene expression datasets. Then, the experimental results of these extended versions of the ML-CFS method will be presented and discussed.

Chapter 6 – The Proposed Multi-Label Correlation-Based Feature Selection Methods Based on Evolutionary Algorithms: This chapter will describe different versions of the ML-CFS method where a Genetic Algorithm (GA) and a lexicographic multi-objective GA were used as the search method, rather than using the simpler hill-climbing search method as in Chapters 4 and 5. The experimental results of the GA-based ML-CFS methods will be presented

and discussed in this Chapter.

Chapter 7 – Conclusions and Future Research: This chapter will present a summary of the contributions of the thesis and also discuss the weaknesses and strengths of the proposed multi-label feature selection methods. Interesting future research directions will also be introduced at the end of this chapter.

1.5 Publications derived from this Research

The research described in this thesis has led to the publication of five peer-reviewed papers, mentioned next in chronological order of publication. The first and fifth papers below were publiched in the proceedings of workshops colocated with international conferences, whilst the other three papers were published in the proceedings of international conferences.

- S. Jungjit, A.A. Freitas, M. Michaelis and J. Cinatl, "A Multi-Label Correlation Based Feature Selection Method for the Classification of Neuroblastoma microarray data", in Advances in Data Mining: 12th Industrial Conference (ICDM 2012): Workshop Proceedings–Workshop on Data Mining in Life Sciences (DMLS 2012), pp. 149–157, I. Bichindaritz, P. Perner, G. Rub, and R. Schmidt, Eds, IBAI Publishing, July 2012.

- S. Jungjit, A.A. Freitas, M. Michaelis and J. Cinatl, "Two Extensions to Multi-Label Correlation-Based Feature Selection: a case study in bioinformatics," in Proceedings of the 2013 IEEE International Conference on Systems, Man and Cybernetics, pp. 1519–1524, Manchester, UK, 2013.

- S. Jungjit, A.A. Freitas, M. Michaelis and J. Cinatl, "Extending Multi-Label Feature Selection with KEGG Pathway Information for Microarray Data Analysis," in Proceedings of the 2014 IEEE International Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB2014), pp. 1–8, Hawaii, USA, 21-25 May 2014.

- S. Jungjit, A.A. Freitas, "A New Genetic Algorithm for Multi-label Correlation Based Feature Selection" In: Proceeding of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, pp. 285–290, 22-14 April, 2015, Bruges, Belgium.

- S. Jungjit, A.A. Freitas, "Lexicographic Genetic Algorithm for Multi-label Correlation Based Feature Selection" In: Proceeding of the Evolutionary Rule-based Machine Learning Workshop: GECCO- Genetic and Evolutionary Computation Conference, pp. 989–996, 11-15 July, 2015, Madrid, Spain.

Chapter 2

Background on Data mining and Bioinformatics

This chapter contains background about knowledge discovery and data mining focusing on the single-label classification task, feature selection for the classification task, and evolutionary algorithms for feature selection. This chapter also briefly describes general background on bioinformatics and molecular biology, in particular on microarray data and other topics relate with our research. The more complex task of multi-label classification will be discussed in Chapter 3.

2.1 Data Mining and Single-Label Classification

Data mining is one of three main phases of the Knowledge Discovery Process (KDP), which aims to discover knowledge/patterns from data in a given application domain. More precisely, as shown in Figure 2.1, the process of knowledge discovery can be divided into 3 broad processes, or phases: (1) Data pre-processing (2) Data mining and (3) Knowledge Post-processing [8, 13, 33, 45, 46, 62, 110]. The third phase is out of the scope of this thesis, so we discuss next the first two phases.

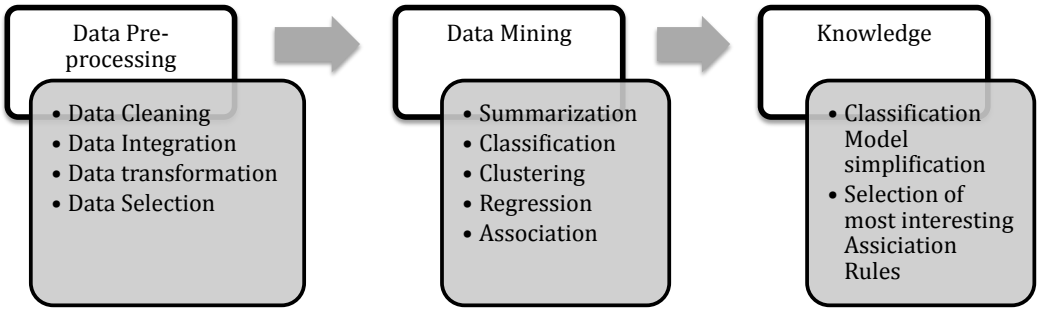


Figure 2.1: The three phases of the Knowledge Discovery Process

Data pre-processing step: The purpose of the data pre-processing phase is transforming raw input data to an appropriate format for the data mining algorithm. Data cleaning deals with noisy, missing values and irrelevant data. Data integration consists of integrating data from a variety of data sources, while data transformation methods consolidate data into an appropriate format before performing data mining. Data selection methods select relevant data for the analysis task. Feature selection is the type of data selection method which aims to selects a relevant feature subset according to an evaluation criterion [76, 78], typically a measure of predictive accuracy in the case of the classification task of data mining, as will be explained later.

Data mining: In this phase, many modeling algorithms can be used according to the target data mining task [17, 109, 116]. The summarization task aims to provide a more compact representation of the data set, including visualization and report generation. Clustering aims to group a set of objects based on their similarity, where a cluster is a collection of similar objects. Data objects in different clusters should have little or nothing in common. Regression consists of finding a function with minimal prediction error to model the data, when the variable to be predicted is continuous (real-valued). Regression analysis is widely used for prediction and forecasting, it is also used to understand which independent variables

(or features) are related to the dependent variable (the variable to be predicted), and to explore the forms of these relationships. Association consists of looking for association relationships or correlation between variables or objects. Typically, associations are expressed in the rule form, showing attribute – values that occur together frequently. Classification is a type of data mining task which aims to learn the relationship between the values of the predictor attributes (or features) of an instance and its class label. This relationship is learned (in the form of a classification model) from pre-classified instances in the training set, and then the learned classification model is used to predict the class label of previously unseen instances in the test set. Note that in classification the class variable takes nominal values (labels) unlike regression, where the predicted variable takes continuous values.

Traditionally, the classification task is defined as a single-label classification problem, where each instance in the data set is associated with just one class label. However, this research addresses a more difficult type of classification problem, namely multi-label classification, as discussed in Chapter 3.

2.2 Single-Label Feature Selection for Classification

Feature selection is a process which selects a relevant feature subset according to an evaluation criterion [19, 20, 39, 54, 67, 76, 77, 79, 97]. In this work we are interested in feature selection for the classification task of data mining. The main objectives of feature selection are to avoid model overfitting and improve the predictive performance of the model [97]. Additional objectives of feature selection are to eliminate irrelevant features and to reduce the computational time taken by the classification algorithm (which will use only the selected feature subset). However, this reduction in computational time is truly beneficial only if the time

taken to perform feature selection is smaller than the corresponding reduction in the time taken by the classification algorithm applied to the selected features.

Furthermore, in several types of applications, such as microarray data (review in Section 2.5) and text document analysis, the data typically has a very high dimensionality and a very small number of instances. In such cases, feature selection is particularly important and it can significantly decrease the risk of model overfiting [39].

2.2.1 Feature Selection Approaches

Feature selection methods can be separated into 3 approaches; (1) the filter approach, (2) the wrapper approach and (3) the embedded approach [19, 20, 39, 67, 76, 77, 79, 97].

There are two groups of methods following the filter approach: (I) feature ranking-based methods and (II) search-based methods. In general, a feature ranking-based method applies statistical techniques to measure the relevance (broadly speaking, correlation with class attribute) of each feature separately, ranks features according to their relevance and selects the top k features from the ranked list (where k is a predefined number). The drawback of this technique is that it considers only one feature at a time (univariate method) and ignores the correlations between features. One feature that is irrelevant by itself can be significantly informative when considered together with other features [43]. Moreover, it tends to select a redundant feature subset.

Another type of filter approach consists of search-based methods. This type of method considers the relationship between features in a feature subset (being a multivariate method), doing a search in the space of possible feature subsets. Each

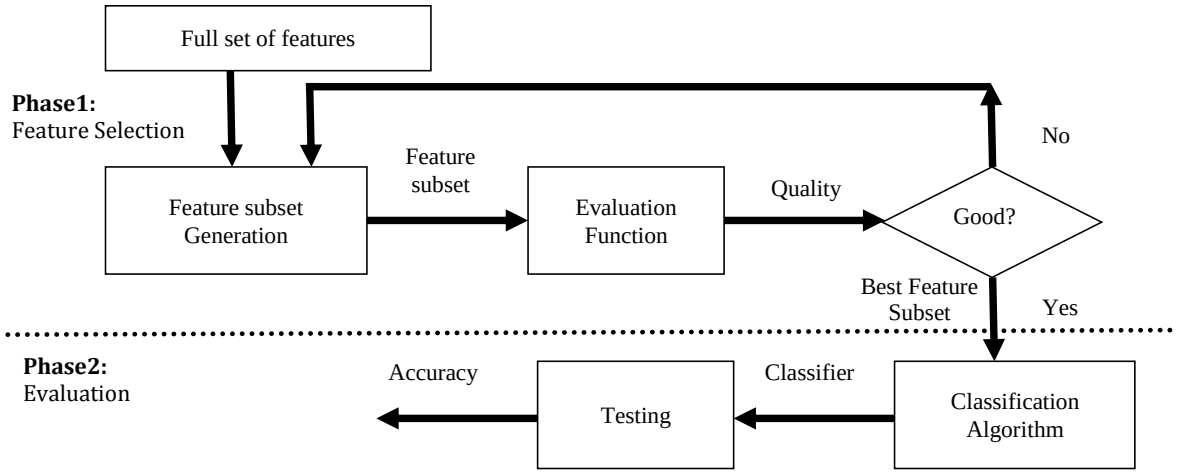


Figure 2.2: The filter approach for feature selection (adapted from [76])

feature subset considered by the search method represents a candidate solution, which is evaluated by an evaluation function (e.g. a correlation-based function). The advantage of this approach is feature redundancy elimination, assuming the evaluation function penalizes redundant feature subsets. On the other hand, in some cases, features with a moderate degree of redundancy are significantly informative when considered together with other features [43].

According to Figure 2.2, in the search-based filter approach, phase 1, the basic flow of feature selection starts with feature subsets which are generated from the full set of features using a search method. Next, each feature subset is evaluated based on a specific criterion (or evaluation function). Both steps in phase 1 are repeated until a stopping criterion is satisfied, e.g. until a fixed number of iterations is performed or the quality of the current best feature subset cannot be improved. Note that all mentioned steps in phase 1 are independent from the classification algorithm, until the system gets the best feature subset. Only in phase 2, executed after we got the best feature subset, the classification algorithm is used. This approach was applied in the design of several feature selection meth-

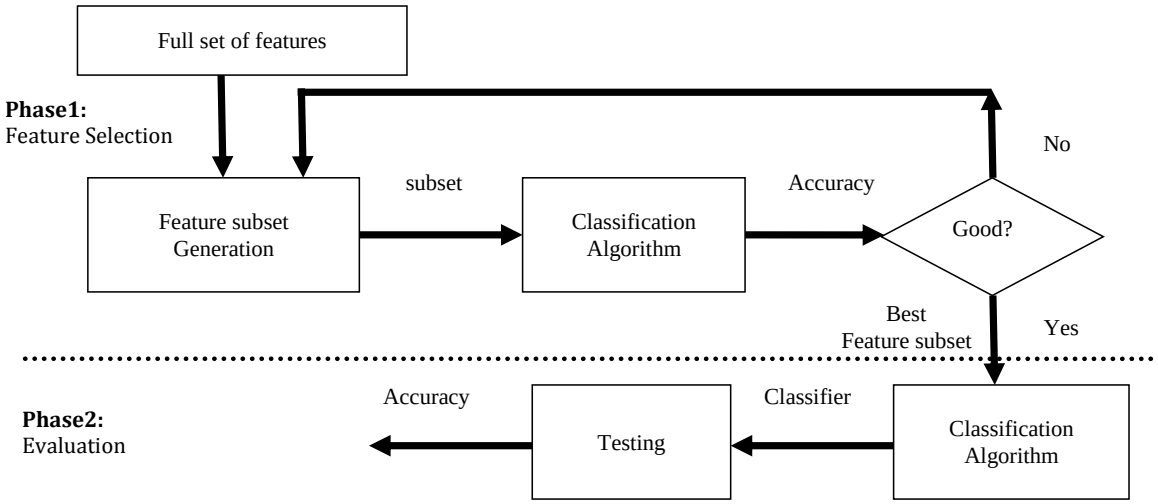


Figure 2.3: The wrapper approach for feature selection (adapted from [76])

ods, such as Correlation-based Feature Selection [44] and Fast Correlation-based Feature Selection [39, 115, 119].

The filter approach is fast, scalable and independent of the classifier. Moreover, [78] highlighted that the most used feature selection approach in real-world applications where the number of features is very large (such as in microarray data and text mining) is the filter approach, because the structure of filter algorithms is simple and it provides a simple way to calculate the relevance of features in large-scale data in a short time.

On the other hand, the wrapper approach selects the best feature subset by doing a search in the feature space guided by a classifier's performance, i.e. using a classifier's accuracy as the evaluation function (Figure 2.3). In the wrapper approach, the classification algorithm used in phase 1 is the same as the algorithm in phase 2, which will use the selected features to build a classifier to be applied to the test set.

The wrapper approach is usually more effective (in terms of maximizing predictive accuracy) than the filter approach because the wrapper approach directly uses the accuracy of the classification model as the evaluation function of a feature subset, but there is a risk of model overfitting [39, 97]. Moreover, the wrapper approach is usually much more computationally expensive than the filter approach because a classification algorithm has to be run for each candidate feature subset, which is not the case in the filter approach.

In the third approach, namely the embedded approach, the search for a good feature subset is embedded into the classifier construction process. Hence, this approach is classifier-specific too, and it also tends to be more computationally expensive than the filter approach. An example of a type of classification algorithm performing embedded feature selection is decision tree algorithms [93], where during the tree construction process, a feature is selected at each internal node of the tree.

Note that both the filter and the wrapper approaches are performed in a preprocessing step, before applying the classification algorithm; whilst the embedded approach is performed as part of the run of a classification algorithm. In this chapter we focus only on feature selection methods performed in a preprocessing phase using the filter approach, i.e., the wrapper and the embedded approach are out of the scope of this work; for the sake of computational efficiency and scalability.

In the context of the filter approach, we can classify feature selection methods into 2 types based on whether or not the method takes into account relationships among features [76]. First, in the univariate filter feature selection approach, the feature selection method measures the quality of just one feature at a time using a given evaluation function, e.g. t-test, F-statistic or information-gain. The advantage of the univariate filter approach is that it fast and scalable [97], but there

are some drawbacks, such as it ignores the dependencies and correlations between features in the feature space.

Second, in the multivariate filter feature selection approach, the feature selection method measures the quality of a feature subset as a whole. That is, the correlation between features in the subset is taken into account. This approach takes more time to generate feature subsets and measure the quality of each feature subset, so it is usually slower than the univatiate approach. Examples of the evaluation functions which are used to measure a feature subset's quality are the correlation-based feature selection (CFS) [44] and Maximize Relevance Minimize Redundant (MRMR)[25, 90]. These evaluation functions will be discussed later in this Chapter.

2.2.2 Feature Selection Methods' Components

A feature selection method consists of two main components: (1) the search strategy and (2) the evaluation function. The first component is a strategy for searching through the space of feature subsets, as discussed next.

2.2.2.1 Search Strategies for Feature Selection

Search strategies can be classified into three broad types: complete, heuristic and stochastic/nondeterministic search [19, 20, 76, 77, 79]

Complete search or exhaustive search evaluates the quality of every candidate feature subset, and returns the best subset. Hence, this method guarantees to find an optimal subset, but its use is not feasible in large-scale datasets, since its time complexity is exponential upon the number of features. Recall that the number of candidate feature subsets is on the order of 2^n, where n is the number of features.

Examples of complete search strategies are the well-known depth-first search and breadth-first search [76].

Whereas complete search strategies are too computationaly expensive in practice, a heuristic search method can find a good solution in a relatively short time, although it risks to miss an optimal solution. Well-known examples of heuristic search methods are best-first search and hill climbing search [76]. The latter was used in [44], and in the first version of our proposed feature selection methods [57], to be described later.

Algorithm 2.1 shows the pseudocode of the best-first search method, adapted from [76]. There are two queues of nodes in the Best-First algorithm: OpenQ is the queue of nodes whose children have not been generated yet (open nodes), and CloseQ is the queue of nodes whose children have already been generated (closed nodes). The algorithm starts by initializing the flag Implement with the value true and adding an initial node to OpenQ. Then Cbest is set to the best node in OpenQ. Next, while the value of the Implement flag is true, the algorithm repeats the following steps. First, it sets QualBest to the quality of Cbest and generates the child nodes of Cbest. Second, it removes Cbest from OpenQ and adds it to CloseQ. Third, for each child node, if that node has not been generated before during the search, it adds that child node to OpenQ and evaluates it. Fourth, it sets Cbest to the best child node. Fifth, if the quality of the best child node is greater than QualBest, it sets the Implement flag to true. Otherwise, it sets Cbest to the best node in OpenQ and checks if the best child node's quality is greater than QualBest. If so, the Implement flag will be set to true, otherwise the Implement flag will be set to false, which will cause the while loop to terminate, and then the algorithm terminates by returning Cbest as the best solution found.

Algorithm 2.1: Pseudocode of Best-First Search()

```
1) Implement = true
2) OpenQ = a queue of nodes whose children have not been generated yet
3) CloseQ = a queue of nodes whose children have been generated
4) Add initial node to OpenQ
5) Cbest = best node in OpenQ
while Implement = True
    do { QualBest = quality of Cbest
         generate child nodes of Cbest
         remove Cbest from OpenQ and add Cbest to CloseQ
         for each child node
           do { if child node has not been generated earlier
                  then { add child node to OpenQ
                         evaluate child node
         Cbest = the best child node
         if Cbest's quality > QualBest
           then Implement = True
           else { Cbest = best node in OpenQ
                  if  Cbest's quality > QualBest
                    then Implement = true
                    else Implement = False
OUTPUT: Cbest
```

Algorithm 2.2: Pseudocode of Hill-Climbing Search()

```
CurrentNode = empty feature subset
CurrentQuality = 0
Implement = true
while Implement = True
do { generate children of CurrentNode
     BestChild = child with the best quality
     BestQuality = quality of the best child
     if BestQuality > CurrentQuality
       then { CurrentNode = BestChild
              CurrentQuality = BestQuality
              Implement = True
       else Implement = False
OUTPUT: CurrentNode
```

Algorithm 2.2 shows the Pseudocode of Hill Climbing search used in [44, 57, 58, 59]. The algorithm starts with the current node representing an empty feature subset and the quality of the current feature subset (CurrentQuality) equal to zero. Also, the flag Implement is set to true. Each iteration of the following while loop performs the following operations. First, the algorithm generates the child nodes from the current node. All child nodes are evaluated using the merit function. After that, the algorithm selects the child node with the best quality, and sets it as the BestChild. Also, the BestQuality is set with the quality of the best child. Next, if BestQuality is greater than CurrentQuality, which means the algorithm found a new child node better than the current node, then CurrentNode is set to BestChild, CurrentQuality is set to BestQuality and the flag Implement is set to True (to make sure the search will continue). Otherwise, the flag Implement is set to false and the while loop terminates. At the end, the algorithm returns the

CurrentNode as the best feature subset found by the hill-climbing search.

Unlike the complete and heuristic search methods mentioned earlier, a nondeterministic strategy searches for good feature subsets using random operators to move in the feature subset space [76] Note, however, that in general a nondeterministic strategy is not completely random, since the application of random operators is guided by an evaluation function. An example of a type of nondeterministic search method is Genetic Algorithms (GAs), which have been extensively used in feature selection [34]. An example of another type of nondeterministic search methods is Simulated Annealing (SA). The pseudocode and flowchart of GA are shown in Algorithm 2.3 and Figure 2.4, respectively, as described below; whilst the pseudocode and flowchart of SA, described further below, are shown in Algorithm 2.4 and Figure 2.5, respectively. More details about GAs for feature selection will be discussed later in this Chapter. Note also that in general non-deterministic methods are heuristic, i.e., they do not guarantee to find an optimal solution.

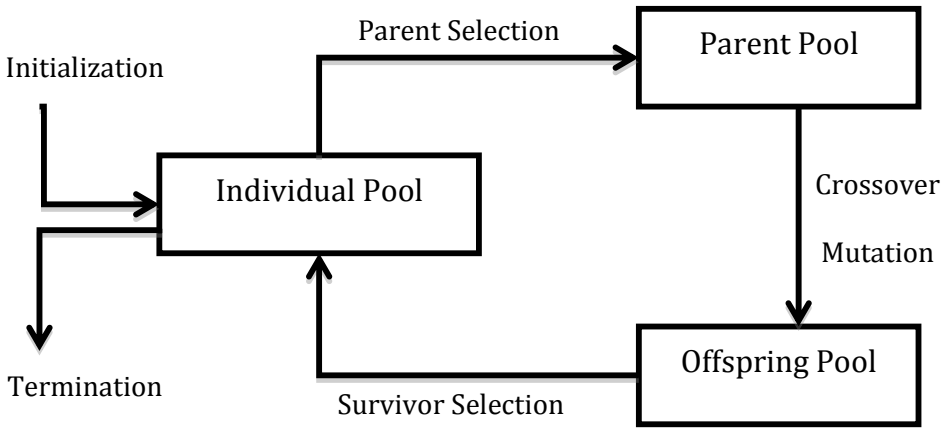


Figure 2.4: General Flowchart of Genetic Algorithms

Algorithm 2.3: Pseudocode of Genetic Algorithms()

```
Initialize candidate solutions
Evaluate each candidate solution
while (condition is not satisfied)
do {
    1) Select parents
    2) Crossover pairs of parents
    3) Mutate the result of Crossover
    4) Evaluate new candidate solutions
    5) Select individual for the next generation
}
```

Genetic Algorithms (GAs) are nondeterministic/random search algorithms based on the evolutionary theory of natural selection and genetics. GAs show a successful exploitation of a random search used to solve optimization problems in many application domains. The main search operators of GAs are inherited from evolutionary theory proposed by Charles Darwin. The key idea of his theory is the "survival of the fittest", which means that the individuals better adapted to their environment will survive in nature, while the rest of them will be vanished with time. As shown as in Algorithm 2.3 and Figure 2.4, first, GAs start with a random individual initialization process which generates a population of individuals – where each inividual is a candidate solution to the target problem. Next, GA selects parent individuals from all individual in an individual pool using some selection approach, e.g. tournament selection or roulette wheel approach. Then, crossover and mutation operations are applied to selected parents, in order to create new individuals. Finally, GA selects survival individuals from offspring individuals and go to next generation.

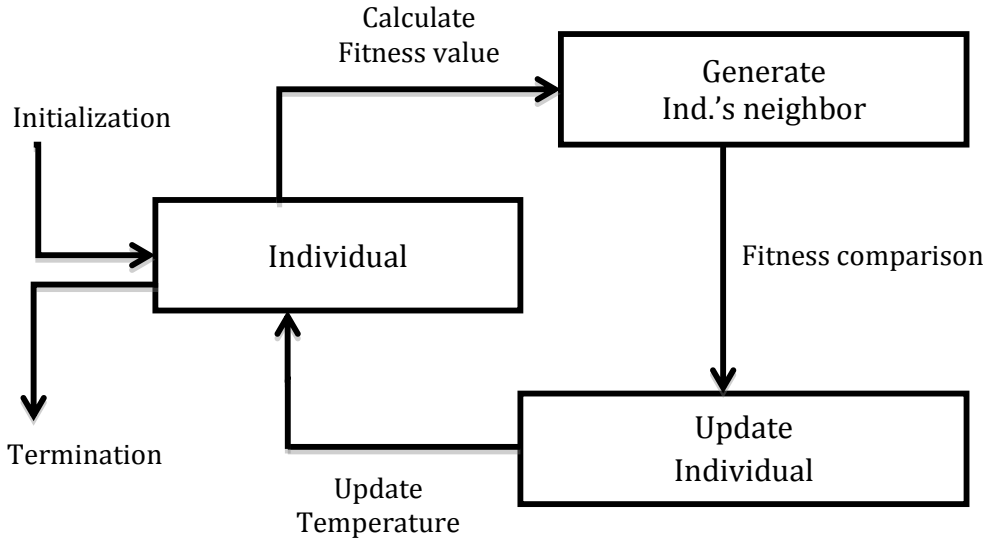


Figure 2.5: General Flowchart of Simulated Annealing Algorithms

---

Algorithm 2.4: Pseudocode of Simulated Annealing()

Initialize a candidate solution X

Set Temperature t = T0

Set Xbest = X

while (condition is not satisfied)

do
- **for** $iteration = 1$ **to** $MaxIteration$
  - **do**
    - S = move X by an operator
    - **if** $f(S) < f(X)$
      - **then**
        - X=S
        - **if** $f(S) < f(Xbest)$ **then** Xbest=S
      - **else**
        - **if** $random < exp(-(f(S) - f(X))/t)$ **then** $X = S$
- t=update (t)

Simulated Annealing (SA) is special variety of hill climbing inspired by the

annealing process in metallurgy [42]. As shown as in the pseudocode of Algorithm 2.4 and in the flowchart of Figure 2.5, first, SA generates a random solution and calculates its fitness (quality) using some fitness function. After that, SA generates a random neighbouring solution and calculates the new solution's fitness. Then, it compares them. If the fitness value of the new solution is smaller (better) than the fitness value of the old solution, then it moves to the new solution. Otherwise, it moves to the new solution with a probability given by the temperature parameter, which increases according to time. This process repeats until an acceptable solution is found or the algorithm reaches some maximum number of iterations. One major disadvantage of SA is the slow convergence speed.

2.2.2.2 Evaluation functions for Feature Selection

The second component of a feature selection method is an evaluation function, which measures the quality of a candidate feature subset based on a predefined criterion; such as the Mutual Information [26, 27, 71]; and Information Gain [74]. In this section, we classify evaluation functions into two main groups: Filter-based evaluation functions and Wrapper-Based evaluation functions as illustrated in Figure 2.6.

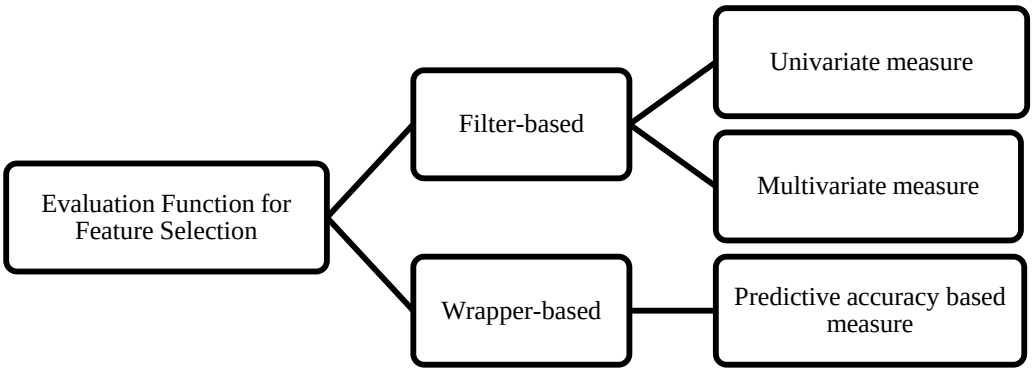


Figure 2.6: Types of Evaluation Function for Feature Selection

In this section we will focus only on fitness functions based on the filter approach, which is the approach followed by the feature selection methods proposed in this thesis, as mentioned earlier. Filter-based evaluation functions can be classified into two main groups: (1) Univariate measures, which evaluate the quality of only one feature at a time; and (2) Multivariate measures, which aim to evaluate a feature subset as a whole, taking feature interaction into account. There are many univariate statistical techniques used to evaluate a feature - e.g. Symmetrical Uncertainty and Information Gain.

Information Gain (IG) is a symmetrical measure used to measure the impurity of a set. In other words, IG measures the amount of information in bits about a random variable Y provided by a random variable X; or equivalently the amount of information about X provided by Y [96]. IG is computed as shown in Equation 2.1,

$$IG = H(Y) - H(Y|X) = H(X) - H(X|Y) \tag{2.1}$$

where $H(X)$ and $H(Y)$ is the entropy of the random variable X and Y, $H(Y|X)$ and $H(X|Y)$ is the conditional entropy of Y given X and of X given Y, respectively.

Mutual information (MI), an information-theoretic measure, was used in many single-label feature selection works [25, 90] for finding the correlation between feature and labels. MI is often used to measure dependencies between nominal variables in feature selection. If the MI between two variables is near zero, this would indicate that the variables are close to independent. The mutual information I(X; Y) between the random variables (feature and class variable) X and Y is shown in Equation 2.2, where p(x,y) denotes the joint probability of feature values x and y, p(x) denotes the marginal probability of x (the probability of the occurrence of event x), the log is in base 2, and the summation is over all values of variables X and Y.

$$MI(X;Y) = \sum \sum p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{2.2}$$

Symmetric uncertainty (SU) is used to calculate the correlation of features and the target class, as shown in Equation 2.3. A feature that has high value of SU is high by correlated with the class variable. There are some benefits of SU, for example, SU balances the bias of mutual information and gives a symmetrical measure for feature correlation by dividing it by the sum of the entropies of X and Y, it reduces the number of feature pairs whose correlations need to be computed (by comparision with MI), since SU(i,j) is the same as SU(j,i); and SU values are normalized. A value 1 of SU(X, Y) indicates that knowledge of one feature's value strongly represents the values of the other feature, and the SU(X, Y) value 0 indicates the independence of X and Y.

$$SU(X,Y) = \frac{2(MI(X,Y))}{H(X) + H(Y)} \tag{2.3}$$

Examples of multivariate evaluation functions which are used to measure a feature subset's quality are Correlation-Based Feature Selection (CFS) and Maximize Relevance Minimize Redundant (MRMR).

Peng et al proposed the MRMR (Max-Relevance and Min-Redundancy) mutual information (MI)-based single-label feature selection method in 2005. This approach aims to find a feature subset which has a high correlation to class labels (high relevance) while the correlation between features in the feature subset is low (low redundancy). They calculate the relevance and redundancy for discrete variables using Equations 2.4 and 2.5; and for continuous variables using Equations 2.6 and 2.7

$$Red(i,j) = \frac{1}{|S|^2} \sum_{i,j \in S} MI(i,j) \tag{2.4}$$

$$Rel(i,h) = \frac{1}{|S|} \sum_{i \in S} MI(i,h) \tag{2.5}$$

$$Red(i,j) = \frac{1}{|S|^2} \sum_{i,j \in S} C(i,j) \tag{2.6}$$

$$Rel(i,h) = \frac{1}{|S|} \sum_{i \in S} F(i,h) \tag{2.7}$$

where $Red(i,j)$ is the redundancy between features i and j
$Rel(i,h)$ is the relevance value of feature i with respect to class h
S is the set of features being evaluated
$MI(i,j)$ is mutual information between features i and j
$F(i,h)$ is the value of the F-statistic between feature i and class h
$C(i,j)$ is the correlation between features i and j

It should be noted that MRMR has the same basic idea as the Correlation-based Feature Selection method (CFS) proposed by [44], in terms of minimizing redundancy between selected features and maximizing the relevance of selected features. The details of the CFS method are described in Section 2.3.

Algorithm 2.5: PSEUDOCODE OF RELIEF()

INPUT: a set of training instances

n = the number of iterations - a user pre-defined number

f = the index of a feature

maxF = the total number of features in the selected instance

OUTPUT: the vector W of estimated feature qualities

SET: all weightsW[f] = 0;

for $i = 1$ **to** n

do {
- Randomly select an instance R_i
- Find nearest hit instance H and nearest miss instance M to R_i
- **for** $f = 1$ **to** $maxF$
  - **do** W[f] :=W[f] - $diff(f, R_i, H)^2 + diff(f, R_i, M)^2$;

A different type of evaluation function based on a multivariate measure of similarity between instances (like in the nearest neighbour classifier) is used by the Relief feature selection method [64]. The main idea of the basic Relief algorithm is to estimate the quality of a feature subset according to how those features distinguish between two instances of different classes that are near to a given instance. The pseudocode of Relief is shown in Algorithm 2.5. First, a randomly selected training instance R_i and its two nearest neighbour instances are selected. Note that one of the two nearest instances has the same class as R_i, called the nearest hit H; and the other nearest instance has the different class, called the nearest miss M. After that, the quality of estimation W[F] is updated for all features of R_i.

If instances R_i and H have different values of the feature f then the feature f separates two instances with the same class, which decreases the quality estimation of f. On the other hand, if instances R_i and M have different values of the attribute f then the attribute f separates two instances with different class

values, which is desirable, so we increase the quality estimation of *f*. The whole process is repeated for n times, where n is a user-defined parameter representing the number of iterations performed by Relief, which is the number of randomly selected instances used to estimate the quality of the feature.

2.3 Single-Label Correlation-Based Feature Selection (CFS)

[44] proposed a feature selection method named Correlation-based Feature Selection (CFS), a well-known filter method for single-label classification. They claimed this method is simple and fast to execute and suitable for both nominal class and continuous class problems (i.e., for both classification and regression problems, respectively). In this research we are interested in this method only in the context of classification problems.

Moreover, Hall stated that a good feature subset should have two main properties: (1) the correlation between each feature and other features in that subset should be low, to minimize feature redundancy; and (2) the correlation between each feature in that subset and the class attribute should be high. In his paper, the merit of a feature subset is evaluated by Equation 2.8:

$$Merit = \frac{k\overline{r_{FL}}}{\sqrt{k + k(k-1)\overline{r_{FF}}}} \tag{2.8}$$

Where ($\overline{r_{FL}}$) is the average feature-label correlation over all feature-label pairs for all features in the current feature subset, ($\overline{r_{FF}}$) is the average feature-feature correlation over all pairs of features in the current feature subset F, and k is the number of features in the current feature subset. In single-label correlation-based feature selection, the quality of a feature subset F depends essentially on two terms, namely ($\overline{r_{FL}}$) and ($\overline{r_{FF}}$). The higher the value of the feature-class corre-

lation and the lower the value of the feature-feature correlation, the higher the quality of the feature subset F with respect to its ability to predict the labels of a single class attribute. In the final experiment of Halls' study, the best-first search method (a popular heuristic search technique) was used for searching the feature subset space [44].

Other study by [119] proposed a fast correlation-based filter approach. Their approach applies Symmetrical Uncertainty (SU), – a measure based on information theory, as a measure to evaluate the correlation between feature-class and feature-feature pairs. The aim of their study was to find the feature subset which is most correlated with the class attribute (according to the SU measure) and which has least redundancy among feature pairs in the feature subset. Therefore, their method is conceptually similar to the correlation-based feature selection method proposed by [44].

2.4 A review of Evolutionary Algorithms for Feature Selection in a Data Preprocessing Phase

Evolutionary Algorithms (EAs) are stochastic (non-deterministic) search methods inspired by the process of natural selection, based on Darwin's evolutionary theory [32]. There are several types of EAs, e.g. Genetic Algorithms, Genetic Programming, and Evolutionary Programming. In this thesis we focus on Genetic Algorithms (GAs), since the vast majority of EAs for feature selection are GAs [34].

The basic principle of GAs as search methods have been discussed earlier in this Chapter; hence, in this Section we discusses GAs specifically in the context of feature selection for the classification task.

In a GA, each individual (candidate solution) is evaluated by a fitness function according to the target problem. In the context of a GA for feature selection (in a data preprocessing phase), an individual is typically represented as a string of bits where each bit takes the value 1 or 0 to indicate whether or not, respectively, a feature is included in the selected feature subset.

In the wrapper approach, the fitness function uses the accuracy of a classification model built with the features selected by the individual, while the filter approach uses a simpler fitness function that is independent from the classification algorithm to evaluate the quality of the feature subset represented by the individual.

A GA for feature selection starts with a population of individuals (candidate feature subsets), and iteratively performs the operations of selecting individuals based on fitness (so that better feature subsets have a higher chance of being selected) and creating new "child" individuals based on variations of the "parent" individuals just selected. This process is iteratively repeated until a stopping criterion (e.g., a fixed number of iterations or generations) is satisfied. Since child individuals tend to inherit characteristics (feature subsets) of good parents (which were selected based on fitness), the population tends to evolve to a near-optimal candidate solution (feature subset). GAs for feature selection have been shown to obtain good predictive accuracy results in single-label classification, by comparison with more traditional search methods often used in feature selection for single-label classification [34, 37, 66, 101].

There are many projects which employed EAs as a feature subset selection method in single-label classification. For instance, [117] proposed the IG-GA approach. This approach is divided into two stages. The first stage is a filtering method, using IG (Information Gain) to calculate the discriminative power of each individual feature (ignoring feature interactions) and selecting the most in-

formative features. The second stage uses a GA as a wrapper method to select, out of all features selected in the filtering stage, a smaller subset of features. They used the K-nearest neighbour method as an evaluator of the IG-GA.

The work by [15] proposed a hybrid IG-GA feature selection method for DNA microarray data. In the first step they calculated a Information Gain-based feature weight for each feature and selected a subset of relevant features based on that criterion. Next, they generated a population for the GA (using features which were obtained from the first step) and evaluated the fitness of an individual based on the accuracy of k-NN. In addition to conventional crossover and mutation operators, the GA uses local search to try to improve candidate solutions.

[110] proposed a different approach to select a feature subset. Their method used multiple evaluation criteria (e.g. t-score, entropy-based and SVM recursive feature elimination) to select a good feature subset in the feature subset space. After that, the best feature subset according to all criteria, overall, was added in a "feature pool" (collection of candidate features). In the next stage, a GA searched for an optimal feature subset from that feature pool, evaluating each individual (candidate fitness subset) using a fitness function based on the classification accuracy and number of selected features.

Next, we present a detailed review of GAs for feature selection in a data preprocessing phase. In general, there are three main components that we need to consider in the design of a GA for feature selection: (1) the individual representation, (2) the fitness function; and (3) the GA operators.

2.4.1 Individual Representation

We can classify individual representations for feature selection into three types: (1) a bit string, (2) a list of feature indexes and (3) a two-part bit string. Most publications used a bit string to represent a candidate solution. The bit string is the simplest individual representation, and it is illustrated in Figure 2.7. As mentioned earlier, the candidate solutions are encoded by a string of n bits where n is the total number of features. The i-th bit with value "1" indicates that the i-th feature was selected, while the i-th bit with value "0" indicates that the i-th feature was not selected. The main drawback of a binary string is the size of the chromosome (or individual) in high dimensional datasets. That is, if the number of genes (n) is very large and we want to represent all genes, we would need a very long chromosome [22]. Moreover, a bit string which has value "1" cannot indicate the level of relevance of the corresponding feature in a chromosome.

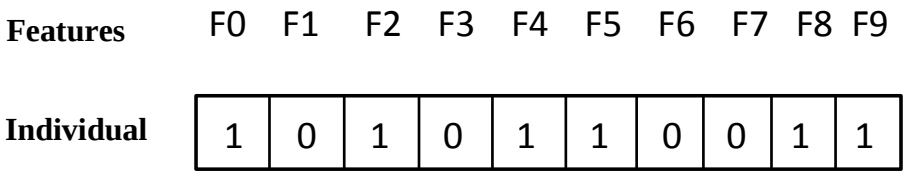


Figure 2.7: Bit String individual representation

Another individual representation is a list of feature indexes [50, 69]. In this case an individual can represent features in two ways: (1) a variable-length list of feature indexes, where each individual consists of at most k feature indexes (where k is a user-defined parameter) and each feature can occur more than once in the list, in different positions, as shown in Figure 2.8. (2) a fixed-length list

of feature indexes, where each chromosome consists of k genes and where each gene represents either the index of a feature or the flag "0" representing no selected feature. The advantage of this representation (in both versions) is that the length of a chromosome does not depend directly on the number of input features. Moreover, the degree of relevance of features can be indicated by the number of occurrences of each feature. That is, a feature with more occurrences in the individual's feature list can be interpreted as a more relevant feature, particularly after many generations of evolution. In terms of crossover effect, if some relevant genes were selected in different positions of the same individual, those genes will have more chance of surviving in both children after performing the crossover operator. However, this representation technique requires a new genetic operator (delete attribute operator), which deletes all copies of a feature index from the chromosome. Furthermore, if one feature occurs in more than one position in an individual, it can act as a redundancy mechanism in the GA (but this redundancy also can indicate relevance, as mentioned earlier).

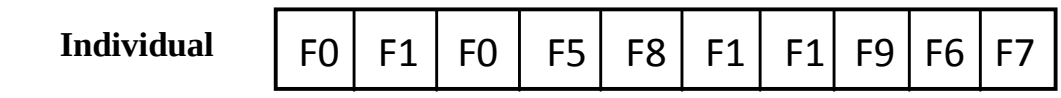


Figure 2.8: A list of feature indexes individual representation

The last one is a two-part bit string representation, where each chromosome represents a candidate solution with additional information on each selected feature. In [47] each individual is separated into two sections, as shown in Figure 2.9: a selected feature section, which is a binary string; and a feature weight section, which is represented by a real-value weight vector (with one weight per feature) for the SVM classifier. The advantage of a two-part bit string is that each chromosome contains both the selected features and other information for the GA or classifier to be built using the selected features. On the other hand, a two-part bit

string requires special crossover and mutation operators, and doubles the size of the chromosome.

A list of the different types of individual representations used by many GAs proposed in the literature is provided in Table 2.1

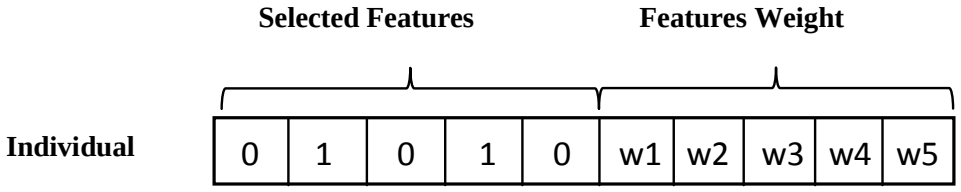


Figure 2.9: A two-part individual representation

2.4.2 Fitness Function

Another component of GAs is a fitness function, which aims to evaluate the fitness of individuals. The vast majority of GAs for feature selection follow the wrapper approach, where the fitness function involves the predictive performance of a classifier built using the features selected by the corresponding individual. However, the filter approach could be used also, without using a classifier's performance [34].

There are several types of feature ranking techniques used in the literature, such as Between Group to Within group sum of square ratio (BW ratio) [15][47], Entropy based [108], Information gain [5, 6, 15], T-statistics [108], the relative approximity degree [82] and Wilcoxon rank sum [75].

A search method using correlation coefficient as the evaluation function [15] and a search for the Markov blanket [125] of the class attribute are examples of a search-based method following the filter approach for feature selection.

In the wrapper approach, the fitness function evaluates candidate solutions

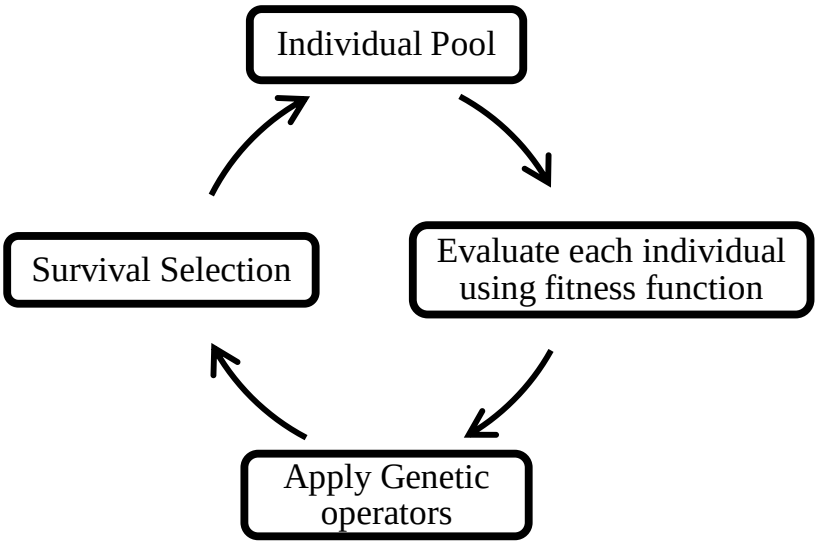


Figure 2.10: General scheme of GAs based on the filter approach

based on the accuracy of a classifier [6, 14, 40, 49, 70, 73, 86]. Some papers use the accuracy of the classifier and another special criterion as a fitness function. For instance, in [70] they use the accuracy of k-NN and the proportion of selected features in the individual to the total number of features in the dataset; in [14] they used the accuracy, the simplicity of decision tree (tree size); and number of features in feature subset; and in [22] they used the accuracy of an SVM and the number of selected features. A list of the different types of fitness functions used by many GAs proposed in the literature is provided in Table 2.1

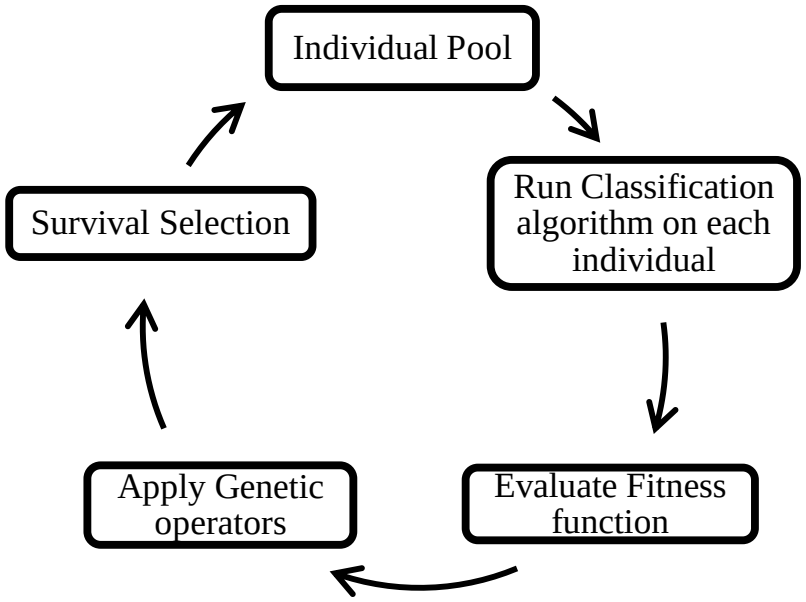


Figure 2.11: General scheme of GAs based on the wrapper approach

Table 2.1: A summary of the literature on Genetic Algorithms for Feature Selection in a data preprocessing phase

References	Feature Selection Approach	Ind.Rep.	Fitness Function	Crossover	Mutation	Other Operation
[69]	Filt & Wrap	List of feature indexes	BW ratio for filter approach The accuracy of k-NN for wrapper approach	Dynamic	Dynamic	Elitist strategy
[5]	Filt & Wrap	Bit string	Information content for filter and The accuracy of Decision Tree, the classification cost for wrapper approach	not mentioned	not mentioned	not mentioned
[70]	Wrap	Bit string	The accuracy of k-NN	Adaptive probability	Adaptive probability	Elitist strategy
[120]	Filt & Wrap	Bit string	PCA for filter approach and the accuracy of MLNB for wrapper approach	Uniform	not mentioned	Elitist strategy
[6]	Wrap	Bit string	The accuracy of Decision Tree and size of the feature subset	not mentioned	not mentioned	not mentioned
[108]	Filt & Wrap	Bit string	Entropy based, T-statistics, SVM-recursive elimination for filter approach and the accuracy of SVM for wrapper approach	Single-point	Bit-flip	not mentioned
[40]	Wrap	Bit string	The accuracy of GRNN	Half uniform	Bit-flip	Simulated Annealing
[14]	Wrap	List of feature indexes	The accuracy and simplicity of Decision Tree	Uniform	Bit-flip	Delete Feature
[86]	Wrap	Bit string	Feature subset cardinality and the accuracy of 1-NN	Multi-point	Bit-flip	Problem–specific operation
[83]	Filt & Wrap	Bit string	The relative proximity degree for filter approach and the accuracy of k-NN for wrapper approach	Multiple-point	Bit-flip	not mentioned
[73]	Wrap	Bit string	The accuracy of SVM	Single-point	Bit-flip	not mentioned
[15]	Filt & Wrap	Bit string	The correlation based feature weights for each feature for filter approach and the accuracy of k-NN for wrapper approach	Standard	Bit-flip	Taguchi method
[22]	Filt & Wrap	Bit string	M Ranked method for filter approach and the accuracy of SVM for wrapper approach	Single-point	Bit-flip	not mentioned
[117]	Filt & Wrap	Bit string	Information Gain for filter approach and the accuracy of k-NN for wrapper approach	Two-point	Bit-flip	not mentioned
[51]	Filt & Wrap	Bit string	Cosine amplitude method and alpha cut method for filter approach and the accuracy of SVM for wrapper approach	One-point	Multi-uniform	Elitist strategy
[75]	Filt & Wrap	Bit string	Wilcoxon rank sum test for filter approach and the accuracy of SVM for wrapper approach	Double one-point	Bit-flip	not mentioned
[50]	Wrapper	List of feature indexes	The accuracy of ANN	One-point	Bit-flip	Speciation, Elitist strategy
[47]	Filt & Wrap	2 parts bit string	BW ratio, the correlation coefficient , the Fisher's discriminant criterion for filter approach and the accuracy of SVM for wrapper approach	Specialized	Specialized	Elitist strategy
[125]	Filt	Bit string	Mainly the generalization error for SVM and feature subset cardinality as a tie-breaking criterion	Standard	Bit-flip	Markov Blanket Based memetic operation

Considering the feature selection approach, most works mentioned in the second column of the table use the filter and wrapper approaches together, in a sequential fashion. The advantage of using the filter approach before applying a GA is the reduction of the number of features in the feature space, in order to allow the subsequent use of a wrapper approach. In contrast, applying only the wrapper approach to all original features would be much more computationally expensive. On the other hand, in works like [125], they do not need to use the filter approach (for feature elimination) because the number of features in the datasets mined in those papers is no more than 100 features, which does not seem too large for a wrapper-based GA for feature selection.

2.4.3 The Main GA Operators: Crossover and Mutation

Another component of GAs is one or more genetic operators which aim to create a new individual(s) from an old one(s). There are two main types of genetic operators: (1) Crossover and (2) Mutation operator. Crossover or recombination merges information from two parents into one or two offspring. There are five main categories of crossover in the literature: One-point crossover, Multi-point crossover, Uniform crossover, Dynamic crossover, and a special crossover.

One-point crossover randomly selects only one crossover position in the parent individuals and swaps gene values to the right of the crossover point between parents producing two children, as illustrated in Figure 2.12. One-point crossover was used in [15, 22, 50, 51, 73, 108].

Multi-point crossover works by first choosing a random number m of crossover points and then the gene values between every two gene sections are swapped between two parents, where a gene section consists of the genes between two successive crossover points. Note that there are two types of multi-point crossover;

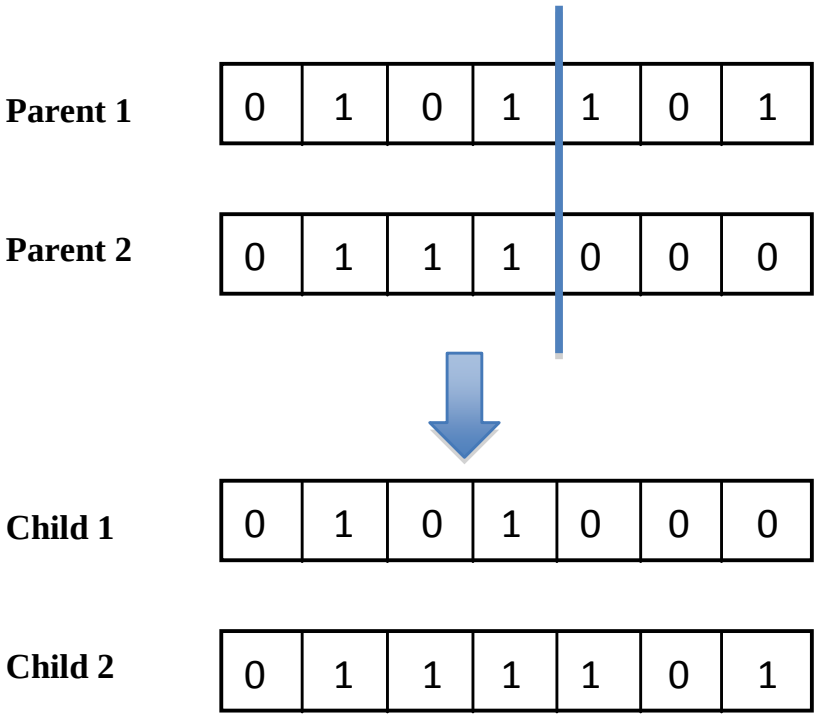


Figure 2.12: One-Point Crossover

odd section swap and even section swap. For example, in even section swap, the section between the first gene and the first crossover point is not swapped, it swaps only the even sections in individuals, e.g. swapping even sections with the genes between the 1st and 2nd crossover points, the 3rd and 4th crossover points, and so on; and vice-versa for odd section swap. An example of even section swap multi-point crossover, with two crossover points, is shown in Figure 2.13. This technique was used in [75, 83, 86, 118] .

Uniform crossover works as follow. First, it generates a string of L random variables between [0, 1], where L is the number of genes. In each position, if the value of that random variable is lower than a pre-defined number p (the probability of crossover per gene), the gene values in this position are swapped between the two parents, to create two children. This type of crossover was used in [14, 120].

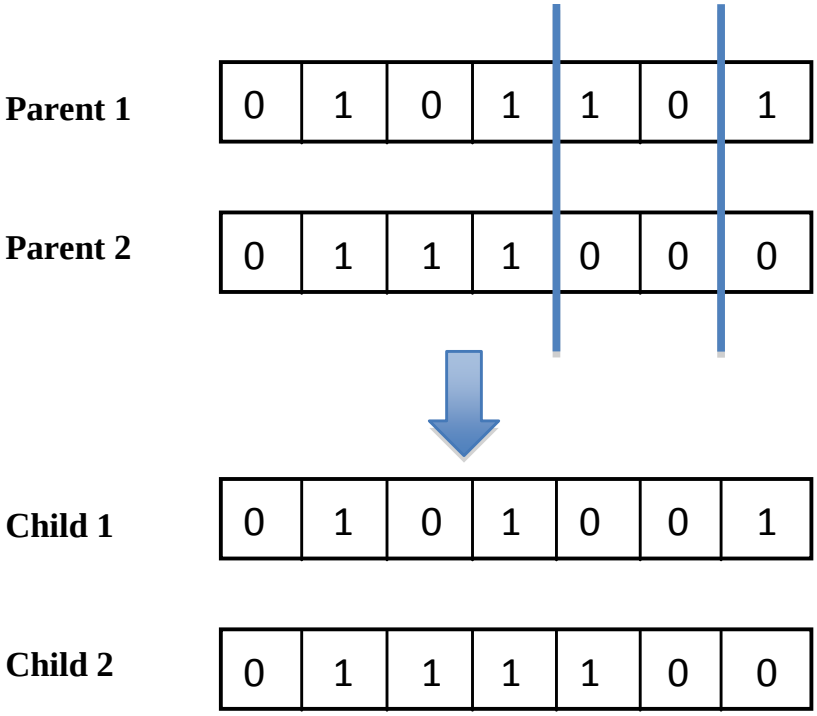


Figure 2.13: m-Point Crossover, m=2

In general, one-point crossover has a high position bias, while uniform crossover tends to have a high distribution bias. In this context, position bias means that, when the GA chooses a crossover point (one point), genes which are close together in an individual are more likely to have their values passed together to children. Note that uniform crossover does not have position bias because the probability of the values of a gene being swapped between the two parents is independent of the position of the gene in an individual.

Uniform crossover has a high distribution bias because the number of swapped genes depended on the probability of crossover per gene, which can be different from 50 %. In [40] another kind of uniform crossover was applied, the half uniform crossover. It calculates the Hamming distance (the number of differing bits)

between the parents. Only half of the different bits of two parents will be swapped.

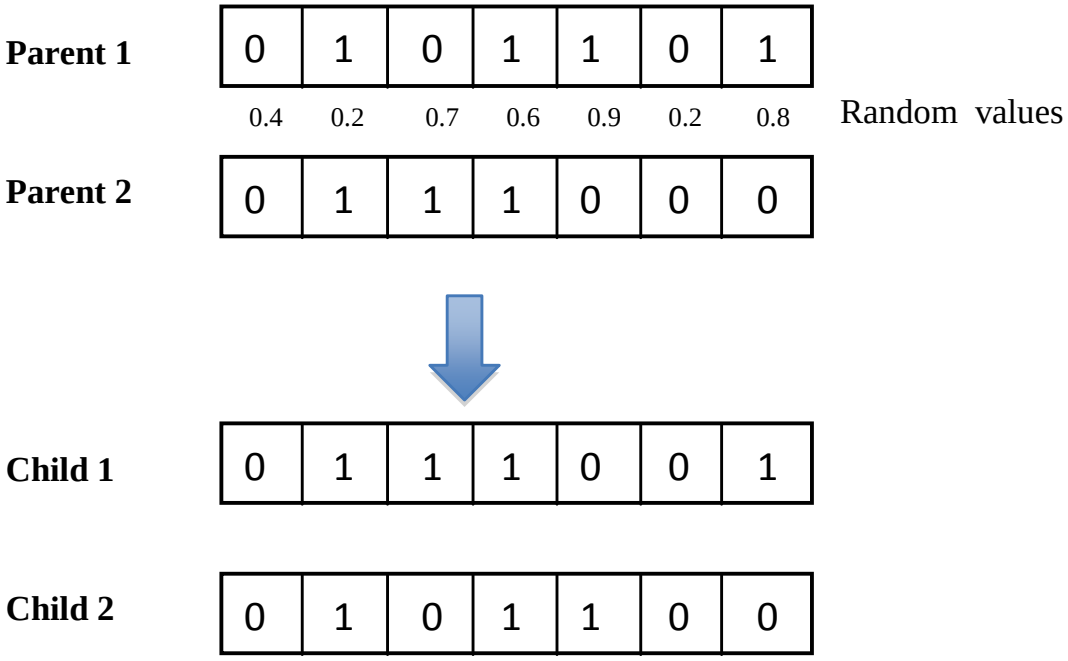


Figure 2.14: Uniform Crossover

Note that GA chooses crossover points (in one-point and multi-point crossover) and gene position (in uniform crossover) without considering the fitness values of individuals.

Dynamic crossover or adaptive crossover was used in [69, 70]. The probability of adaptive crossover is varied depending on the fitness value of solutions [106, 107]. The advantage of adaptive crossover is that it improves the convergence rate of the GA and avoids the GA being trapped in a local minimum.

In [47] researchers use a special crossover operator, which is designed for the two-part bit string representation discussed earlier. In that paper, a special crossover conserves the genes shared by the parents (for the first part of an

individual) and the SVM weight information in the second part of individual.

Another type of genetic operation is mutation, which considers each gene separately and allows each gene to flip (bit-flip mutation) according to the mutation rate (a user-specified parameter). Usually, a large value of mutation rate would lead the GA into a purely random search. To avoid this problem, the mutation rate is usually small, typically in the range of 0.005-0.05 (i.e.,0.5-5%). Most papers on GA for feature selection used bit-flip mutation in their study. Instead of a fixed mutation rate, [69, 70] applied adaptive mutation, where the mutation rate is a dynamic value, which iteratively changes based on the fitness value of parent chromosomes. Those papers claim that using adaptive mutation and crossover can balance the capacity of exploitation and exploration of GA.

The list of the different types of crossover and mutation operators used by many GAs for feature selection proposed in the literature is provided in the fifth and sixth columns of Table 2.1, respectively.

2.4.4 Other Operations

Finally, most papers also use other operations/techniques to increase the performance of GAs such as an elitist strategy and speciation strategy. The elitist strategy, which is used in [47, 51, 69, 70, 120], aims to preserve the best individuals, which have the highest fitness values, for the next generation (without performing any genetic operation on those individuals). It is used in GAs to make sure the best individual survive and to guarantee that the best fitness value of each generation would not be worse than the one in the previous generation.

In addition, [50] used another technique called Speciation, which is used to identify a species of solutions within the population. In general, this technique

uses a distance function to evaluate similarity between two solutions (individuals) in the population. If the similarity value of those individuals is too low they are considered to belong to different species and the crossover operator does not operate between those individuals. The speciation is promising for obtaining diverse solutions of high accuracy.

Other specific techniques are a delete feature operator which uses only one parent to produce a child in [14], Taguchis' method which is used in [15] for improving local search in GA and a Markov Blanket based operation for removing or adding features in a feature subset in [125].

2.5 Background on Gene Expression from a Bioinformatics Perspective

The development of microarray technology has lead to a new direction of biological research, and provided a new type of problem for machine learning research. The small glass chip or gene expression microarray is used to measure the gene expression levels in tissue samples. Gene expression levels can distinguish among groups of patients' tissue conditions, and help physicians to diagnose whether a patient has disease or not. Microarray technology was developed for measuring the gene expression levels of tens of thousands of gene simultaneously. Surely, the main challenge for machine learning or data mining algorithms is the dimensionality of the data (the number of genes), which is very high compared to the typically very small number of samples (instances) [80]. In this section, we will first describe general background on bioinformatics such as basic concepts of gene expression and microarray data. Next, the challenge of microarray data for data mining will be described.

2.5.1 Gene Expression

Gene expression is a biological process which converts the information encoded in genes into proteins. Genes are contained in DNA (Deoxyribonucleic Acid) strains. The basic flow of sequential transformation, where DNA is transformed to proteins, can be separated into two stages: (1) DNA transcription, in this stage DNA is transcribed to mRNA (messenger Ribonucleic Acid) and (2) Translation stage, in this stage RNA is translated to protein [31, 80].

According to Figure 2.15, the process of protein synthesis starts with the DNA replication process. In this stage, a DNA strain is replicated from one strain to two strains. After that, DNA strain transcript information is coded into a temporary molecule, called mRNA. Finally, the protein is built using sequential information (sequential order of amino acids) from mRNA in the translation stage.

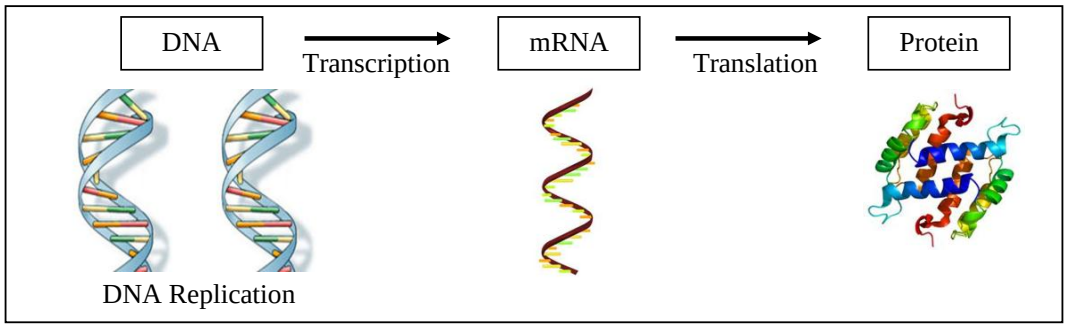


Figure 2.15: Sequential process for protein synthesis from DNA

2.5.2 Characteristics of DNA Microarray

DNA microarray is a chip-based technology which is widely used to study biomedical samples [31]. A microarray may contain thousands of spots; each spot on the chip represents a different coding sequence from different genes. In a microarray

experiment, the example tissue is grown in two different conditions (a reference condition and a test condition). Next, RNA is extracted from the two cells, and is labelled with different dyes (red and green) during the synthesis of cDNA (complementary DNA) by reverse transcriptase. Note that cDNA is a double-stranded DNA synthesized from RNA. After that, cDNA is hybridized onto the microarray slide. The microarray slide is placed inside a dark box where it is scanned with a laser at suitable wavelengths to detect the red and green dyes. Finally, the result of hybridization is stored as a file for further analysis [4].

Instances	**Gene 1**	**Gene 2**	**Gene 3**	**...**	**Gene *n***	**Class**
1						
2						
...						
m						

(a)

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{a_{11}} & \cdots & \boldsymbol{a_{1n}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{a_{m1}} & \cdots & \boldsymbol{a_{mn}} \end{bmatrix}$$

(b)

Figure 2.16: Two types of data structure to store microarray data: (a) table or (b) matrix

According to Figure 2.16, data from microarray can be stored in a table or matrix where each row represents a data instance or sample (or cell line in the case of our experiments reported later), and each column represents a feature or attribute (corresponding to a gene). The cell i,j in the matrix is the gene expression value of gene j in instance i., i=1,...,m; j=1,...,n, where m is the number of instances and n is the number of genes.

2.5.3 The Challenge of Microarray Data for Data Mining

The main challenge of microarray data related to the data mining area is its high dimensionality; the number of features (genes) is very large, typically many thousand genes; while the number of instances is small, typically a few tens of instances [31, 39]. Microarray data usually has tens of thousands of genes (features), while it often has only a few tens of samples (instances). The problems are that the large number of features (genes) can lead to high computational time for the data mining algorithm and most features are irrelevant or very noisy, potentially leading to an overfitting of the classification model to the data as explained later – note that in this research we focus on the classification task of data mining. Due to the very high dimensionality of microarray data, it is desirable to employ some data mining methods which can select informative feature subsets in microarray data.

Hence, the feature selection methods proposed in this thesis can be used to select features in microarray data. This will be shown in Chapter 5, where we will propose feature selection methods tailored to exploiting biological background knowledge and evaluate those methods in microarray datasets.

2.6 Summary

This Chapter presented background on data mining and bioinformatics, focusing on conventional single-label feature selection for the classification task. First, filter and wrapper feature selection approach were described in Section 2.2.1. In addition, this chapter described the feature selection algorithms' components: (1) the search method, and (2) the evaluation (fitness) function. In particular, it reviewed both hill-climbing based and evolutionary algorithm-based methods for feature selection in Section 2.2.2.1.

Section 2.2.2.2 described the evaluation function component of feature selection methods. Evaluation functions based on the filter approach; such as Mutual Information, Information Gain, and Symmetric uncertainty were reviewed. Also, examples of multivariate evaluation functions such as the correlation-based feature selection (CFS) and Maximize Relevance Minimize Redundant (MRMR) were described.

Moreover, Section 2.4 reviewed evolutionary algorithms for feature selection, including: (1) individual representation, (2) fitness function, and (3) the main evolutionary search operators. Last, general background on bioinformatics and molecular biology, especially on microarray data and other topics related with our research, was briefly reviewed in Section 2.5.

Chapter 3

Multi-Label Classification and Feature Selection

In this chapter we present the background on multi-label classification problems, multi-label feature selection methods, and multi-label classification algorithms. We also discuss several multi-label classification evaluation measures.

3.1 Multi-Label Classification Problems

Multi-label classification is different from traditional single-label classification because in multi-label classification each instance can be associated with a set of class labels, while in traditional single-label classification each instance is associated with only one class label. For example, an article about social media can be classified to both information technology and social activity class labels. A document can be classified to the class labels education and economics at the same time. A gene can be associated with many biological functions in an organism, and an image can be annotated with both sea and mountain class labels.

Multi-label classification is used in many areas; such as, text classification, scene classification, music classification, bioinformatics and medical diagnosis [113]. The basic idea of multi-label classification is illustrated in Table 3.1, where each

Table 3.1: An example of multi-label data set

InstanceID	Class 1	Class 2	Class 3
1	X		
2			X
3	X	X	X
4		X	
5		X	X

instance can belong to more than one category (class label).

Predicting class labels for multi-label classification problem is more complicated when compared with traditional single-label classification problems. Generally, multi-label classification methods can be classified into two groups: (1) problem transformation methods, which transform a multi-label classification problem into single-label classification problems and predict each class label separately using a single-label classification algorithm; and (2) algorithm adaptation methods, which modify a single label classification algorithm to perform multi-label classification directly [24, 112].

3.2 Multi-Label Problem Transformation Methods

In this section we review the main methods that transform a multi-label classification problem to one or more single-label classification problems [24].

Firstly, some problem transformation methods transform a multi-label classification problem to just one single-label classification problem, such as the dubbed PT1 method or Label Elimination method, which randomly selects one of the multiple labels of each multi-label instance and discards the other labels of that instance. PT1 is illustrated in Table 3.2, which shows a possible result of applying PT1 to the data in Table 3.1. It is also possible to select the labels to be discarded

Table 3.2: Transformed data using PT1

InstanceID	Class 1	Class 2	Class 3
1	X		
2			X
3	X		
4		X	
5			X

Table 3.3: Transformed data using PT2

InstanceID	Class 1	Class 2	Class 3
1	X		
2			X
4		X	

from each multi-label instance using a non-random criterion such as selecting the label with maximum or minimum frequency in the dataset [10, 92].

Other method, dubbed PT2 (also called Instance Elimination method), removes all instances which have multiple labels from the dataset and uses the remaining instances for data mining. Table 3.3 shows the result of applying PT2 to the data in Table 3.1. A clear weakness of both PT1 and PT2 is that they lead to an information loss, because these techniques tend to eliminate lots of data from the original data set. [94, 95] applied these methods in their research.

The PT3 method or Label Power set method also proposed in [112], which creates a single label for each element in the power set of the set of labels (i.e., for each possible combination of labels) that is observed in the dataset. This method does not lead to information loss like PT1 and PT2, but PT3 can lead to a large number of class labels. This is a serious problem particularly when the number of instances is small, since in this case there would be too few instances for some class labels, making it very difficult to reliably predict those labels. This technique is used in [95, 111, 114]. A variation of PT3 is the pruned transformation method, which was proposed by [94]. This method prunes away label sets that occur a

Table 3.4: Transformed data using PT3

InstanceID	Class 1	Class 2	Class 3	Class 2 & Class 3	Class 1& Class2 & Class 3
1	X				
2			X		
3					X
4		X			
5				X	

Table 3.5: Transformed data using PT4

InstanceID	Class1	¬Class1
1	X	
2		X
3	X	
4		X
5		X
InstanceID	Class2	¬Class2
1		X
2		X
3	X	
4	X	
5	X	
InstanceID	Class3	¬Class3
1		X
2	X	
3	X	
4		X
5	X	

number of times smaller than a small user predefined threshold. The result of applying the PT3 method to the data in Table 3.1 is shown in Table 3.4.

PT4, also call Binary Relevance, is a method which transforms the original data set into $|L|$ new data sets (where L is set of labels). Each data set contains all data instances of the original dataset. In the *i-th* dataset, $i = 1,\ldots, |L|$, each instance is assigned a single label, which is i if the instance contained the *i-th* label in the original dataset, and $\neg i$ otherwise. This technique is used in [23] and [111]. Table 3.5 shows the result of applying PT4 to the data in Table 3.1. Note that

Table 3.6: Transformed data using PT5

InstanceID	Class
1	Class1
2	Class2
3	Class1
3	Class2
3	Class3
4	Class2
5	Class2
5	Class3

PT4 creates three single-label datasets, so three classifiers need to be trained.

The last problem transformation method is PT5. This method decomposes each instance into n rows (where n is the number of true labels for the current instance), where those rows have the same attribute values but different classes. However, this method leads to a large amount of data replication in the dataset. The result of applying PT5 to the data in Table 3.1 is shown Table 3.6. Note that PT5 creates a dataset where some instances are duplicated with respect to the features, differing only in their class labels. This would be a problem for most conventional classification algorithms, so this method is rarely used in practice.

The second group of multi-label classification methods consists of algorithm adaptation methods. These methods modify a conventional single-label classification algorithm to solve a multi-label classification problem. Some of these methods are briefly discussed in Subsection 3.3. In any case, note that these methods are not the focus of this research (which focuses on data preprocessing methods).

A similar taxonomy, using somewhat different terminology was introduced in [21], who classified multi-label classification methods into two main types: (1) algorithm independent and (2) algorithm dependent. Algorithm independent methods correspond to the problem transformation method proposed by [112]. Algo-

Table 3.7: A comparison of problem transformation methods proposed or discussed in different works.

Methods	Advantages	Disadvantages	Number of classifiers	Number of instances
1) PT1: Label elimination	Simple and easy to implement	Information loss	one	Same as for original data set
2) PT2: Instance elimination	Simple and easy to implement	Information loss	one	Reduced
3) PT3: Label creation or Label power set (LP)	Considers some relationship between labels	A large increase in the number of class labels, increasing the risk of model overfitting	one	Same as for original data set
4) PT4: Label based transformation or Binary Relevance (BR)	Simple and easy to implement	Considers each label separately, ignoring label correlations; slow (leads to many runs of a classification algorithm)	Increased: Equal to the number of labels	Increased in total (over all new data sets)
5) PT5	Simple and easy to implement	create duplicated instances regarding to feature values and instances with inconsistent class	one	Increased

rithm independent methods can be used with any type of classification algorithm, whereas algorithm dependent methods use a specific type of algorithm for dealing with multi-label classification problems.

Table 3.7 shows a comparative study of problem transformation methods proposed or discussed by different authors. For each method, the first column mentions its name, the second and third columns mention the advantage(s) and disadvantage(s) of that method, the fourth column indicates the effect of using the method on the number of single-label classifiers that need to be trained after the data has been transformed, and the fifth column indicates the effect of using the method on the number of instances in the data being mined.

3.3 Multi-Label Classification Algorithms

Several single-label classification algorithms have been modified for multi-label classification. For example, the C4.5 algorithm (a well-known decision tree induction algorithm proposed in [93]) was modified by [16]. In order to extend C4.5 to

a multi-label scenario, Clair and King adapted the formula of entropy calculation for multi-label classification. In this Section we focus only on the two multi-label classification algorithms used in our experiments reported in Chapter 4-6, namely multi-label extensions of the kNN (K nearest neighbour) and Backpropagation Neural Network algorithms. Each of these is described in a separate subsection, in the following.

3.3.1 Multi-Label K-Nearest Neighbours Algorithm

A multi-label classification algorithm based on an extension of a traditional single-label k-nearest neighbours (kNN) algorithm, called ML-kNN was proposed [124]. This algorithm works as follows. For each test instance unseen, ML-kNN identifies that instance's k nearest neighbours in the training set and considers which of those neighbours are labelled as positive or negative. Next, in order to transfer class labels from those neighbours to that unseen instance, in essence, this approach uses the k-NN algorithm independently for each label in the label set. More specifically, it counts the number of neighbours associated with each label and uses a maximum a posteriori principle to define the label set for the unseen instance.

For an unknown-class instance x, the predicted value (0 or 1) of each class label Y_j is computed by Equation 3.1,

$$Y_j = \begin{cases} 1, if P(c_j|Y_j = 1)P(Y_j = 1) \geq P(c_j|Y_j = 0)P(Y_j = 0) \\ 0, otherwise \end{cases} \tag{3.1}$$

where:

c_j is the count of the nearest neighbours of instance x which have the j-th label (i.e., nearest neighbours with $Y_j = 1$), $P(c_j|y_j = 1)$ is the probability of the count value c_j conditioned on the event that instance x has the j-th label, $P(c_j|Y_j = 0)$ is

the analogous probability conditioned on the event that x does not have the j-th label, $P(Y_j = 1)$ and $P(Y_j = 0)$ are the prior probability of the j-th label taking the value 1 or 0 (estimated by taking into account the relative frequency of $y_j = 1$ and $y_j = 0$ in the entire training set).

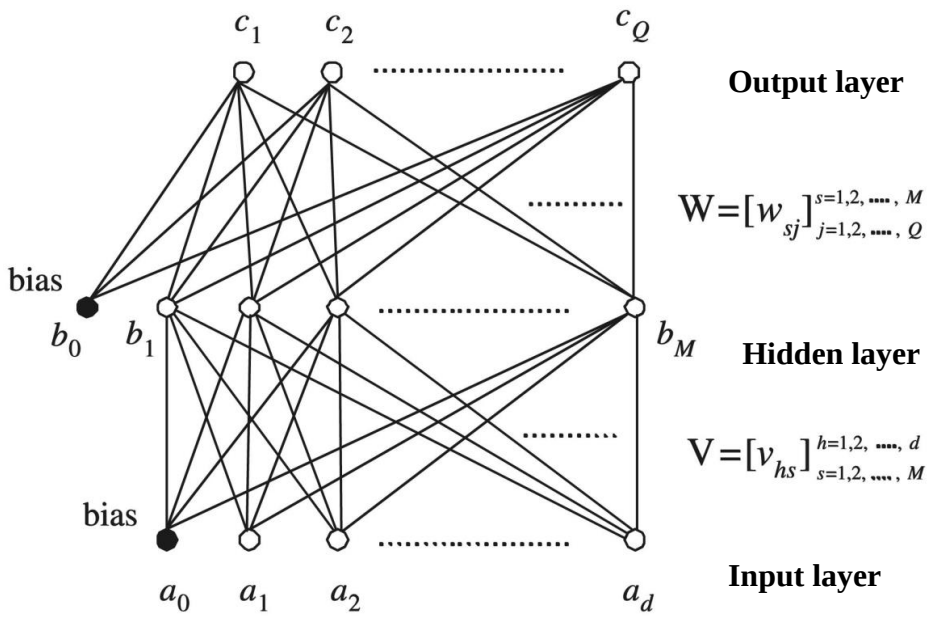
The ML-kNN method was used in multi-label classification of music into emotions [111], multi-label classification for video annotation [23] and multi-label learning with label-specific features [121]. Moreover, [111] pointed out that ML-kNN is a high performance representative of problem adaptation methods.

An aspect of the original single-label kNN which is inherited to ML-kNN is the distance measure. For distance-based classification methods like kNN using the Euclidean distance measure, feature normalization is an important step, because it prevents a feature with initially (before normalization) large range from outweighing a feature with initially smaller range when computing the distance between two instances. Feature normalization equalizes the range of values of all features [2, 72, 100]. The Euclidean distance is used to measure the distance between instances in ML-kNN; therefore, the original features need to be normalized in a pre-processing process, before the application of ML-kNN.

3.3.2 Multi-Label Neural Network Algorithm

An extension of the traditional feed-forward neural network for multi-label classification problem, Backpropagation Multi-Label Learning (BPMLL), was proposed by [123]. A feed-forward neural network has a multi-layer architecture. The first layer represents an input layer and the last layer is the output of the algorithm. Layers in the middle, called hidden layers have no connection with the external world. Each layer has many neurons (nodes), which connect to all nodes in the next layer, while there is no connection between nodes in the same layer. Note

that the output layer has one node for each of the class labels.



Where

Y is the set of class labels
d is the number of input nodes (the dimensionality of the feature vector)
Q is the number of output nodes, each corresponding to one of the possible class labels
M is the number of nodes in the hidden layer
V_{hs} is the weight of the connection between input node h and hidden node s, $(1 \leq h \leq d, 1 \leq s \leq M)$
W_{sj} is the weight of the connection between input node j and hidden node s, $(1 \leq j \leq Q, 1 \leq s \leq M)$
$a_0, \ldots, a_d$ are the input nodes (a_0 is the bias node)
$b_0, \ldots, b_M$ are the hidden nodes (b_0 is the bias node)
$c_0, \ldots, c_Q$ are the output nodes (representing class labels)

Figure 3.1: Backpropagation Multi-Label Learning (BPMLL) architecture (adapted from [123])

This kind of architecture is show in Figure 3.1. There are d units in the input layer each one corresponding to each feature while there are q unit of output layer where each unit corresponding to class label.

The neural network is trained with the gradient descendent and error back

propagation with an error function. The global error function is shown in Equation 3.2. The error term for the i-th instance is calculated as the accumulated difference between the output of each pair for nodes where one node (c_k^i) represents a label belonging to instance i and another node (c_l^i) represents a label not belonging to instance i. Note that the bigger the difference ($c_k^i - c_l^i$), the better the predictive performance of the neural network, since the output of c_k^i should be as high as possible (label k occurs in instance i) and the output of c_l^i should be as low as possible (label l does not occur in instance i).

$$E = \sum_{i=1}^{m} E_i = \sum_{i=1}^{m} \frac{1}{|Y_i||\overline{Y_i}|} \sum_{(k,l)\in Y_i \times \overline{Y_i}} exp(-(c_k^i - c_l^i)) \tag{3.2}$$

where

Y_i is the set of labels occurring in the instance i

$\overline{Y_i}$ is the complementary set of Y_i (i.e., set of labels not occurring in instance i)

$c_k^i - c_l^i$ is the difference between the output of the node for one label belonging to instance i ($k \in Y_i$) and one label not belonging to instance i ($l \in \overline{Y_i}$)

k is the index of a label belonging to label set Y_i

l is the index of a label belonging to label set $\overline{Y_i}$

m is the number of instances in a multi-label training set

In Equation 3.2, the larger the value of $c_k^i - c_l^i$, the smaller the value of $exp(-(c_k^i - c_l^i))$, and so the smaller the error associated with the pair of labels k and l. The summation of these errors for each pair of labels is then normalized by dividing that summation by the total number of label pairs ($|Y_i||\overline{Y_i}|$), for each instance i, and finally the errors for all instances are added up to calculate the global error E.

3.4 Multi-Label Feature Selection Methods

There are a small number of published studies on filter-based feature selection methods for multi-label classification following the data preprocessing (rather than the embedded) approach, as follows.

Several works first transform the multi-label problem into a single label problem and then use a single-label feature selection method. In [26] proposed to use a problem transformation method to transform data from a multi-label problem to a single-label problem, and used the mutual information (MI) as an evaluation function for feature subset selection in the filter approach. The Pruned Problem Transformation Method (PPT) is a variation of the Power Set problem transformation method (PT3) defined in [112], which simply considers each different label subset in the original data as a single label (as in PT3 method) and removes from the data set the new labels with a number of instances smaller than a predefined threshold. Then they used greedy forward feature selection based on MI to select features. Similarly, in [27] the PPT method was applied for transforming data and multivariate mutual information was used to select features. This paper claims that using multivariate mutual information can deal with redundancy between features in the feature subset. However, these studies cannot deal with multi-label problems directly.

RF-BR used the binary relevance (BR) transformation technique to transform multi-label data to single-label data and then evaluated each feature subset using ReliefF (RF). This approach also cannot directly deal with multi-label datasets [103].

The main drawback of using a problem transformation method in those studies is that they cannot cope with the correlation between labels. Other multi-label feature selection methods which avoid to use a problem transformation method

were proposed in several studies, as follows.

Multivariate mutual information for multi-label feature selection without using problem transformation was proposed by [71]. This approach avoids the information loss during the problem transformation process. However, this approach needs a user pre-defined number (the number of features in the selected feature subset), which equals to three in their paper.

In [68], authors modified the idea from the fast correlation-based feature selection (FCFS) method which was proposed by [119] and applied it in a multi-label scenario. They used maximum spanning tree (MST) and symmetrical uncertainty (SU) in their filter approach to select features in a multi-label classification task. They built a SU matrix which considers feature-feature correlations and feature-label correlations using SU as a criterion to measure correlations. However, they assumed all features were discrete, a drawback in datasets where many features are continuous. Continuous features can be discretized in a preprocessing step, but this leads to loss of relevant information, especially in microarray datasets with more than 20,000 continuous features such as the data used in our experiments reported in Chapter 5.

A multi-label feature selection method using an MF-statistic and MreliefF based approach was proposed by [65]. These two approaches take the label correlation into account by using the multi-label F-statistic and multi-label reliefF method to evaluate the correlation between a feature and labels, but they cannot consider the correlation between features.

Also, [120] performed feature selection for classification with multi-label naive Bayes. First they used Principle Component Analysis (PCA) to remove redundant features, and after that they used a Genetic Algorithm (GA) for selecting a relevant feature subset. In their paper the learning problem was addressed by

multi-label naive Bayes (MLNB). Their study performed feature selection in a multi-label scenario because the GA uses the predictive performance of MLNB to guide the search for features, following a wrapper approach. However, note that PCA is an unsupervised learning method for dimensionality reduction, whereas the datasets used in our experiments are appropriate for supervised learning methods. In addition, PCA creates new features that are difficult to be interpreted by users, whilst a dimensionality reduction approach based on feature selection has the advantage of preserving the meaning of the original features, facilitating the user's interpretation of the classifier built with the selected features [97] [76].

Relief for multi-label feature selection (RF-ML) was proposed by Spolaor and others in 2013. This approach searches for k nearest multi-label instances by using a dissimilarity function. RF-ML considers the effect of feature interaction when computing the dissimilarity between instances. The dissimilarity function used in their paper is the normalization of Hamming Distance. Another method proposed by [103], IG-ML selects feature subsets which have a multi-label information gain (IG) value greater than or equal to a pre-defined threshold. This method has the drawback of requiring an ad-hoc user-defined threshold value.

In [74], authors proposed the multi-label feature selection via information gain (IGMF). This approach evaluates the information gain between a feature and the label set and after that eliminates irrelevant features (using the average of the information gain across all features as a threshold). They claim that this approach can deal with the multi-label problem directly. However, a discretization technique was used before calculating information gain, and as mentioned earlier this involves information loss especially in datasets with many continuous features.

Also, [91] adopted the information gain-based feature selection for multi-label scenario. This approach computes a multi-label information gain score for all features then ranks all features before selecting the top k features, where k is a

Table 3.8: A Summary of work on Filter-based Multi-Label Feature Selection Methods

Ref.	**PT Method**	**Eval. Function**	**Disadvantages**
[26, 27]	Power set	MI	• Cannot deal with multi-label problem directly • Loss of information associated with discretized data • Need user pre-defined number of selected features
[65]	None	F-statistic	• Ignore the correlations between pairs of features • Need user pre-defined number of selected features
[71]	None	MMI	• Loss of information associated with discretized data • Need user pre-defined number of selected features
[68]	None	SU	• Loss of information associated with discretized data • Need user pre-defined number of selected features
[103]	BR	ReliefF	• Need user pre-defined number of selected features • Cannot deal with multi-label problem directly
[74, 91, 103]	None	IG	• Need user pre-defined number of selected features • Loss of information associated with discretized data
[105]	None	RFML	• Need user pre-defined number of selected features
[104]	BR	IG	• Need user pre-defined number selected features • Loss of information associated with discretized data

user-defined parameter.

Another method is proposed by [104]. The main idea of this approach is to deal with label dependency. This method constructs a new label from an original label pair for q times (while q is a user-predefined number, the number of constructed labels, where q is smaller than the total number of labels). After generating the q new labels then BR was applied to a new dataset which consists of the original dataset plus q constructed labels. The main drawbacks of this approach is that it needs a user-predefined number (q), also, there are many ways to generate a new label (by using AND, XOR or XNOR operator) and the user needs to specify how to select a pair of labels. Moreover, this approach increases the number of labels in the dataset regarding to the size of q.

Table 3.8 shows a summary of the previously discussed feature selection methods based on the filter approach.

3.5 Multi-Label Classification Evaluation Measures

In multi-label classification, the predictive accuracy measures are different from conventional measures for single-label classification. The main measurers for evaluating multi-label predictive accuracy are Hamming-loss, Ranking-loss, One-error, Coverage, Precision, Recall, Exact Match, F-measure and Accuracy [113]. These measures are described below.

3.5.1 Hamming-Loss

Hamming Loss is an evaluation function which takes into account prediction errors (an incorrect label is predicted) and omission errors (a label is not predicted). The Hamming loss is defined as:

$$HammingLoss = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Y_1 \triangle Z_i|}{|L|} \tag{3.3}$$

Where

D is a multi-label test data set, consisting of $|D|$ multi-label instances $(x_i, Y_i), i = 1..|D|$

Y_i is the set of class labels associated with the i-th instance, $Y_i \subseteq L$

L is the set of class labels

$|L|$ is the number of labels in L.

Z_i is the set of labels predicted by the multi-label classifier for the i-th instance

$\triangle$ is the symmetric difference of two sets and corresponds to the XOR operation in Boolean logic. That is, a class label belongs to the set of labels defined by $Y_i \triangle Z_i$ if and only if that label occurs in either Y_i or Z_i, but not in both sets.

3.5.2 Ranking loss

Ranking Loss is an evaluation function which expresses the number of times that irrelevant labels are ranked higher (better) than relevant labels, averaged over all instances in the test data set. A label is said to be relevant (irrelevant) for an instance if that instance has (does not have) that label. For each instance, labels are ranked in decreasing order of their probability of belonging to that instance, as estimated by the classification algorithm.

$$RankingLoss = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{|Y_i||\overline{Y_i}|}|\left\{(y_1,y_2)|f(x_i,y_1)\leq f(x_i,y_2),(y_1,y_2)\in Y_i\times\overline{Y}_l)\right\}| \tag{3.4}$$

Where

Y_i is the set of true labels of instance x_i

$\overline{Y_i}$ is the complementary set of Y_i with respect to the label set L

$|D|$ is the number of instances in the test data set

y_1 and y_2 are a pair of relevant and irrelevant labels for x_i respectively

x_i is the set of predictor attribute values in the i-th instance

$f(x_i, y_i)$ is the score of label y_i in instance x_i computed by the multi-label classifier (the higher score, the higher probability of x_i being associated with label y_j)

3.5.3 One Error

One error evaluates how many times the top-ranked label is not in the set of relevant (true) labels of the instance. The top-ranked label for an instance is the label with the highest estimated probability of belonging to that instance, and the one error value is averaged over all test instances.

$$OneError = \frac{1}{|D|}\sum_{i=1}^{|D|}\delta(arg\min_{\lambda\epsilon Y_i} r_i(\lambda)) \tag{3.5}$$

Where

$|D|$ is the number of instances in the test data set

λ is a label belong to the label set L

$\delta(\lambda) = 1$ if $\lambda \notin Y_i$, 0 otherwise

$r_i(\lambda)$ is the ranking of label λ in the i-th instance

Y_i is the set of labels associated with the i-th instance

3.5.4 Coverage

Coverage evaluates how far we need to go down the ranked list of labels (in decreasing order of label probability as estimated by the multi-label classifier) in order to cover all the relevant (true) labels of an instance, averaged over all test instances.

$$Cov = \frac{1}{|D|} \sum_{i=1}^{|D|} \max_{\lambda \epsilon Y_i} r_i(\lambda) \tag{3.6}$$

Where

$|D|$ is the number of instances in the test data set

$r_i(\lambda)$ is the ranking of label λ in the i-th instance

3.5.5 Precision

Precision evaluates the proportion of relevant (true) labels that are selected over the set of predicted labels for an instance, averaged over all test instances.

$$Precision = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Z_i|} \tag{3.7}$$

Where

D is a multi-label test data set, consisting of $|D|$ multi-label instances $(x_i, Y_i), i = 1..|D|$

x_i is the set of predictor attribute values in the i-th instance

Y_i is the set of class labels associated with the i-th instance. $Y_i \subseteq L$

L is the set of class labels

Z_i is the set of labels predicted by the multi-label classifier for the i-th instance

3.5.6 Recall

Recall evaluates the proportion of relevant (true) labels that are selected over the set of true class labels associated with an instance, averaged over all test instances.

$$Recall = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Y_i|} \tag{3.8}$$

Where

D is a multi-label test data set, consisting of $|D|$ multi-label instances $(x_i, Y_i), i = 1..|D|$

x_i is the set of predictor attribute values in the i-th instance

Y_i is the set of class labels associated with the i-th instance. $Y_i \subseteq L$

L is the set of class labels

Z_i is the set of labels predicted by the multi-label classifier for the i-th instance

3.5.7 Exact Match

Exact Match calculates only fully correct predictions of all class labels.

$$EM = \frac{1}{|D|} \sum_{i=1}^{|D|} I(Y_i = Z_i) \tag{3.9}$$

Where

D is a multi-label test data set, consisting of $|D|$ multi-label instances $(x_i, Y_i), i = 1..|D|$

x_i is the set of predictor attribute values in the i-th instance

Y_i is the set of class labels associated with the i-th instance. $Y_i \subseteq L$

L is the set of class labels

Z_i is the set of labels predicted by the multi-label classifier for the i-th instance

$I(\cdot)$ is the indicator function, that return 1 if its argument is true and 0 otherwise.

3.5.8 Accuracy

Accuracy evaluates the proportion of relevant (true) labels that are predicted over the total number of predicted or actual labels for an instance, averaged over all test instances.

$$Precision = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{|Y_i \cap Z_i|}{|Y_i \cup Z_i|} \tag{3.10}$$

Where

D is a multi-label test data set, consisting of $|D|$ multi-label instances $(x_i, Y_i), i = 1..|D|$

x_i is the set of predictor attribute values in the i-th instance

Y_i is the set of class labels associated with the i-th instance, $Y_i \subseteq L$

L is the set of class labels

Z_i is the set of labels predicted by the multi-label classifier for the i-th instance

3.5.9 F-measure

F-measure evaluates the proportion of relevant (true) labels that are predicted over the summation of the number of predicted and actual labels for an instance, averaged over all test instances. It combines the ideas of prediction and recall into a single formula.

$$F - measure = \frac{1}{|D|} \sum_{i=1}^{|D|} \frac{2 \times |Y_i \cap Z_i|}{|Y_i| + |Z_i|} \tag{3.11}$$

Where

D is a multi-label test data set, consisting of $|D|$ multi-label instances $(x_i, Y_i), i = 1..|D|$

x_i is the set of predictor attribute values in the i-th instance

Y_i is the set of class labels associated with the i-th instance. $Y_i \subseteq L$

L is the set of class labels

Z_i is the set of labels predicted by the multi-label classifier for the i-th instance

3.5.10 Summary of Multi-Label Predictive Accuracy Measures

For each evaluation measure, the prediction of class labels for an instance is measured from a "fully correct" or "partly correct" perspective (depending on the level of correctness). Also, the multi-label evaluation method can be classified from different perspectives into example-based, label-based and a ranking-based measures. Label-based measures compute some measure for predictive accuracy separately for each label and average the results. Such measures are out of the scope of our experiments. We focus instead on example-based and ranking-based measures, some of which are used in our experiments reported in Chapter 4-6.

Table 3.9 shows the summary of all evaluation measures from different perspectives. Only one measure, namely the Exact Match measure, takes into account only fully correct predictions and completely ignores partly corrected predictions. The rest of the measures take both fully correct and partly correct perspectives into account.

From a multi-label classification point of view, no single predictive accuracy measure is enough to capture different aspects of multi-label classification, due to the complexity of multi-label classification [18, 112]. Hence, five different popular

Table 3.9: A summary of multi-label predictive accuracy measures from different perspectives

Measure Type	**Fully correct**	**Partly correct**
Example-Based	Exact Match	Hamming-Loss, Accuracy, Precision,Recall, F-measure
Ranking-Based	None	One-Error, Coverage, Ranking-loss,Average Precision,

measures of multi-label predictive accuracy were used in our experiment: Average Precision (Avg.Pre), which is to be maximized, while Coverage (Cov.), Hamming Loss (H.Loss), One-error (One-Err) and Ranking Loss (R.Loss) are to be minimized. These measures were used because they seem the most used ones in the literature, and represent a good diversity of perspectives to evaluate multi-label predictive accuracy.

3.6 Summary

This Chapter has reviewed the main concepts and methods for multi-label feature selection. First, the multi-label classification problem and problem transformation methods were introduced. The differences among problem transformation methods were discussed in Section 3.2. Then, two well-known multi-label classification algorithms which are used in our experiments in Chapters 4 through 6 were described (see Section 3.3). Next, in Section 3.4, the advantages and disadvantages of the filter-based multi-label feature selection methods proposed in the literature were reviewed and discussed. Finally, a number of well-known multi-label classification evaluation measures were reviewed in Section 3.5.

Chapter 4

New Multi-Label Correlation-Based Feature Selection Methods Based on Hill Climbing Search

This chapter describes several versions of the proposed Multi-Label Correlation-based Feature Selection (ML-CFS) method [58] based on hill climbing search. This method extends the single-label CFS method to the more complex multi-label classification scenario. We first describe the first version of the ML-CFS method in Section 4.1, and then describe two different generic extensions of this method [58][59] in Section 4.2. These extensions are generic in the sense of being independent of the application domain of the data being mined. By contrast, Chapter 5 will present ML-CFS extensions specifically designed for biological datasets. In Section 4.3 we describe the datasets used in the experiments. In Section 4.4 we report computational results comparing the first version of ML-CFS and the two generic extensions of ML-CFS. In Section 4.5 we report results comparing ML-CFS with baseline multi-label feature selection methods. A general discussion of the results will be presented in Section 4.6.

4.1 The First Version of the Multi-Label Correlation-Based Feature Selection (ML-CFS) Method

The essential idea of the multi-label CFS method is to extend the evaluation function of the single-label CFS method proposed in [44]. Recall that, in single-label CFS, the evaluation function is used to measure the quality (merit) of a candidate feature subset using Equation 4.1, where $\overline{rFL}$ is the average feature-label correlation over all feature-label pairs and $\overline{rFF}$ is the average feature-feature correlation over all pairs of features, F is the candidate feature subset being evaluated, L is the set of class labels, and k is the number of features in F.

$$Merit = \frac{k\overline{r_{FL}}}{\sqrt{k + k(k-1)\overline{r_{FF}}}} \tag{4.1}$$

$$\overline{r_{FF}} = \frac{\sum_{f_i=1, f_j=1, i \neq j}^{|F|} r_{f_i f_j}}{f_p} \tag{4.2}$$

$$r_{f\overline{L}} = \frac{\sum_{i=1}^{|L|} r_{fL_i}}{|L|} \tag{4.3}$$

$$\overline{r_{FL}} = \frac{\sum_{f=1}^{|F|} r_{f\overline{L}}}{|F|} \tag{4.4}$$

In the multi-label approach, like in the single-label approach [44], we use Equation (4.2) to estimate the term $\overline{rFF}$. This is because, for a given dataset, both the single-label and the multi-label problems use the same set of features.

In order to compute $\overline{rFF}$, Equation (4.2) is computed for each pair of features f_i and f_j in the dataset, and then the results are averaged dividing the total summation of all results by the number of pairs of features, denoted *fp* in Equation(4.2).

The difference between the conventional single-label approach and our multi-label approach is in the way that the term $\overline{rFL}$ is estimated. The basic idea is that

we calculate the average feature-label correlation using the arithmetic mean of all feature-label pairs (i.e., the average value of the correlation coefficient between each feature in a candidate subset F and each label in the set of all class labels) by using Equation (4.3) and (4.4). By contrast, in the conventional single-label CFS method, the computation of $\overline{rFL}$ is substantially simpler, requiring only the mean of the correlation between each feature in F and the single class attribute - i.e, using only Equation (4.4).

The Pearson's correlation coefficient (r) between two continuous variables x and y is shown in Equation (4.5).

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})\sum_{i=1}^{n}(y_i - \overline{y})}} \tag{4.5}$$

Where
x_i and $\overline{x}$ are the value of variable x in the i-th instance and the average value of x, y_i and $\overline{y}$ are the value of variable y in the i-th instance and the average value of y; and n is the number of instances in the training set.

Another important component of a feature selection method is the search approach, which is used for creating the candidate feature subsets to be evaluated. Hill-Climbing search, a well-known heuristic search approach [76], was used in the proposed ML-CFS method. Heuristic search can find a good solution in a relatively short time, although it risks to lose an optimal solution. However, in practice heuristic methods are needed because the size of the search space (i.e., the number of candidate feature subsets) grows exponentially with the number of features.

ML-CFS' pseudocode, shown in Algorithm 4.1, works as follows. Firstly, we set the merit of the current feature subset (Curr-Merit) to -1, and set the merit of

the best feature subset (Best-Merit) to 0. The former two variables are used for checking the termination criterion at each iteration of the while loop. Then the best feature subsets (Best-Feat-SS) and the current feature subset (Curr-Feat-SS) are initialized with the empty set. Secondly, we compare the best merit with the current merit. If Best-Merit is smaller than or equal to Curr-Merit then the while loop will stop, otherwise the system will update the values of the variables containing the current feature subset's merit, the current feature subset and the feature subset list. Note that at the start of the first iteration of the while in iteration loop the condition (Best-Merit $>$ Curr-Merit) is true, since those two variables were initialised with 0 and -1, respectively; but at the start of each of the other while loop iterations that condition will be true (so that the loop proceeds) only if the previous iteration was successful in finding a new feature subset which had a better merit than the previously known Best-Merit.

Next, for each feature f which is not in the current feature subset (Curr-Feat-SS), we create a new feature subset (New-Feat-SS) by computing the union of the current feature subset with feature f, and then we calculate the merit of the new feature subset (New-Feat-SS) using Equation (4.1). After that, we update the feature subset list (Feat-SS-List) by adding the pair of new feature subset and new merit into the list.

Finally, we select the new feature subset with the highest value of new merit in the feature subset list and assign that feature subset to the best feature subset (Best-Feat-SS) and its merit to the best merit (Best-Merit) in this while loop's iteration. After that, we move to the next iteration, and so on, until the loop's exit condition is satisfied.

Algorithm 4.1: ML-CFS WITH HILL-CLIMBING SEARCH()

```
Set Curr-Merit = -1 /* Merit of the current feature subset */
Set Best-Merit = 0 /* Merit of the best feature subset found so far */
Set Best-Feat-SS = empty set /* Best feature subset produced so far */
Set Curr-Feat-SS = empty set /* Current feature subset */
while (Best-Merit > Curr-Merit)
do {
    Set Curr-Merit = Best-Merit
    Set Curr-Feat-SS = Best-Feat-SS
    Feat-SS-List = [ ]
    for each feature f not in Curr-Feat-SS
    do {
        Set New-Feat-SS = Curr-Feat-SS ∪ f
        Set New-Merit = Merit of New-Feat-SS, measured by Eq. (4.1)
        Add pair (New-Feat-SS, New-Merit) to Feat-SS-List
    }
    Select the New-Feat-SS with highest value of New-Merit in Feat-SS-List
    Set Best-Feat-SS = selected New-Feat-SS
    Set Best-Merit = selected New-Merit
}
```

4.2 Two Generic Extensions of the ML-CFS Method

The two extensions of ML-CFS described in this Section are generic in the sense that they are independent of the type of dataset being mined. By contrast, Chapter 5 will describe three extensions of ML-CFS that exploit biological knowledge, and were designed to be used in biological datasets.

4.2.1 ML-CFS Using the Absolute Value of Correlation Coefficient

In the original multi-label ML-CFS method described in Section 4.1 and the original single-label CFS method [8], Pearson's linear correlation coefficient (r) was used to estimate the terms $\overline{rFF}$ and $\overline{rFL}$ in Equation (4.1). In general, there are two types of correlation: positive correlation and negative correlation. Both of them can represent redundancy between a pair of features, or represent the relevance of a feature to predict a set of labels, as follows. For the purpose of measuring redundancy between two features, what matters is the absolute value of the correlation coefficient (r), regardless of its sign. E.g., both $r = +0.8$ and $r= -0.8$ represent a strong degree of redundancy. However, in the original single-label and multi-label CFS methods, the values of the merit formulas depend on both the value and the sign of r. If a feature subset contains, say, one pair of features with $r = +0.8$ and another pair of features with $r = -0.8$, these two values would cancel each other resulting in an average r over those two feature pairs of 0; a misleading value, since the two r values actually suggest a large degree of redundancy in each of those feature pairs.

To avoid the aforementioned problems, we use the absolute (without sign) value of the correlation coefficient in all occurrences of the correlation coefficient r in Equation (4.1) when calculating the value of the average correlation between features in a feature subset F ($\overline{rFF}$) and the average correlation between features and labels ($\overline{rFL}$). Hence, the average correlation between features in a feature subset F ($\overline{rFF}$) is computed by Equation (4.6), where *fp* is the number of feature pairs in feature subset F. The average value of the correlation coefficient between features and labels is given by Equation (4.7), which uses Equation(4.8) to compute the average value of the correlation coefficient between each single feature and all labels. Note that $r_{f_i f_j}$ and r_{fL}return a value in [0..1].

$$\overline{r_{FF}} = \frac{\sum_{f_i=1, f_j=1, i \neq j}^{|F|} \left| r_{f_i f_j} \right|}{f_p} \tag{4.6}$$

$$\overline{r_{FL}} = \frac{\sum_{f=1}^{|F|} \left| r_{f\overline{L}} \right|}{|F|} \tag{4.7}$$

$$r_{f\overline{L}} = \frac{\sum_{i=1}^{|L|} |r_{fL_i}|}{|L|} \tag{4.8}$$

4.2.2 ML-CFS Using Mutual Information for Class Label Weighting

In the original ML-CFS method, Equation (4.3) computes, for a given feature *f*, the arithmetic average of the correlation between that feature and a class label over all labels, implicitly assuming that all labels are equally relevant and ignoring dependencies between labels. However, in real-world datasets there might be a significant degree of dependence between some labels, where the occurrence of one label would increase the probability of another label for a given instance. For example, in multi-label classification of emotions in a music dataset, the class label 'Sadness' might be more correlated with the class label 'Depressing' than with the class label 'Cheerful'. The correlation between labels is important in multi-label classification [122]. If the labels were independent from each other, we could simply transform a multi-label problem into a set of single-label problems using the binary relevance method. However, when there are strong dependences among labels in the data, simply using an approach that ignores label correlations, like binary relevance or computing the arithmetic average of correlations across all labels may not be sufficient to cope well with the label-dependence problem.

To take label dependences into account, we used mutual information (MI) to measure the degree of dependence between each pair of labels. We use MI, rather than Pearson's correlation coefficient, because labels are nominal, rather than nu-

merical, and MI is often used to measure dependencies between nominal variables in feature selection [71][65][26]. If the MI between two variables is near zero, this would indicate that the variables are close to independent.

The mutual information MI(X;Y) between the random variables (class attributes) X and Y is shown in Equation (4.9), where p(x,y) denotes the joint probability of class labels x and y, p(x) denotes the marginal probability of x, the log is in base 2, and the summation is over all values of variables X and Y. To use MI as a measure of label dependence, we first compute the average MI of each label L_i (AvgMI(L_i)) as defined in Equation (4.10). This is simply the mean of the MI between label L_i and each of the other class labels L_j (j $\neq$ i).

$$MI(X;Y) = \sum \sum p(x,y) log \frac{p(x,y)}{p(x)p(y)} \tag{4.9}$$

$$AvgMI(L_i) = \frac{\sum_{i=1,j\neq i}^{|L|} MI(L_i L_j)}{|L| - 1} \tag{4.10}$$

The AvgMI(L_i) value for each label L_i can then be used to modify the Merit function as follows. When computing the correlation between a feature and a set of labels, Equation (4.3) is extended by assigning a different weight to each feature-label correlation term (for each label L_i), where the weights are based on the *AvgMI* values computed by Equation (4.10). We investigated two opposite approaches to assign such weights, based on two opposite rationales, as follows.

On one hand, it could be argued that a greater weight should be assigned to feature-label correlations involving labels with greater *AvgMI* values. The rationale for this is that, if a given label L_i is highly correlated with the other labels – i.e., $AvgMI(L_i)$ is large, one should reward features which are strong predictors of that label because a multi-label classification algorithm exploiting label correlations could use an accurate prediction of that label to improve the accuracy in the prediction of other labels. Hence, one approach investigated in this work is to

Table 4.1: Main Characteristics of the Datasets used in the experiments

Dataset Symbol	Dataset Name	Dataset Description					
		Instances	Features	Labels	Label Cardinality	Label Density	Distinct Labels
N1	CAL500	502	68	174	26.044	0.150	502
N2	Scene	2407	294	6	1.074	0.179	15
N3	Emotions	593	72	6	1.869	0.311	27
N4	Yeast	2417	103	14	4.237	0.303	198
N5	Business	11314	21924	30	1.600	0.053	158
N6	Art	7484	23146	26	1.659	0.063	404
N7	Education	12030	27534	33	1.455	0.044	348
N8	Recreation	12828	30324	22	1.428	0.065	369
N9	Health	9205	30635	32	1.635	0.051	235
N10	Entertainment	12730	32001	21	1.405	0.067	246
N11	Computer	12444	34096	33	1.518	0.046	296
N12	Science	6428	37187	40	1.471	0.037	332
B1	Enron	1702	1001	53	3.378	0.064	753
B2	Medical	978	1449	45	1.245	0.028	94

extend Equation (4.3) with Equation (4.11).

$$r_{f\overline{L}} = \frac{\sum_{i=1}^{|L|} |r_{fL_i}| \, AvgMI(L_i)}{\sum_{i=1}^{|L|} AvgMI(L_i)} \tag{4.11}$$

$$r_{f\overline{L}} = \frac{\sum_{i=1}^{|L|} |r_{fL_i}| \, (1 - AvgMI(L_i))}{\sum_{i=1}^{|L|} (1 - AvgMI(L_i))} \tag{4.12}$$

On the other hand, it could be argued that a greater weight should be assigned to feature-label correlations involving labels with smaller $AvgMI$ values. The rationale for this is that, if a given label L_i is weakly correlated with the other labels – i.e., $AvgMI(L_i)$ is small, a multi-label classification algorithm exploiting label correlations would not be able to use an accurate prediction of other labels to improve the accuracy in the prediction of label L_i, and therefore features which are strong predictors of that label should be rewarded regardless of their ability to predict other labels. Hence, one approach investigated in this work is to extend Equation (4.3) with Equation (4.12) In Equations (4.11) and (4.12), the denominators normalize the weight values so that the sum of weights across labels is 1.

4.3 Datasets Used in the Experiments

Table 4.1 shows the main characteristics of all the multi-label datasets used in our experiments. There are two different groups of datasets based on the data type of their features and their application domain: (1) N1-N12, multi-label datasets with continuous (real-valued) features; and (2) B1-B2, multi-label datasets with binary features. The datasets in Table 4.1 were obtained from MULAN repository [`http://mulan.sourceforge.net/datasets.html`].

The datasets are described in Table 4.1. In this table, the titles of the first five columns have self-explanatory meanings. The meanings of the last three columns are as follows.

Label Cardinality (LCard) is the average number of labels per instance. Label Density (LDen) is the label cardinality divided by the number of labels. Distinct Labels (DistL) is the total number of distinct label combinations observed in the dataset [112]. The formal definitions of Label Cardinality, Label Density and Distinct Label are as follows.

$$LCard = \frac{1}{|D|}\sum_{i=1}^{|D|}|Y_i| \tag{4.13}$$

$$LDen = \frac{LCard}{L} \tag{4.14}$$

$$DistL = |Y_i \subseteq L|\exists(x_i, Y_i) \in D| \tag{4.15}$$

where $|D|$ is the number of instances in dataset D, Y_i is the set of class labels occurring in the i-th instance, L is the set of class labels and (x_i, y_i) denotes the i-th instance's feature set and label set.

4.3.1 Pre-processing of the Multi-Label Datasets

Zero-mean normalization was used to normalize all features in all datasets consisting of continuous features (N1-N12). I.e., a feature's mean value is normalized to 0, and the value of a feature for an instance was normalized to the number of standard deviations above or below the feature's mean. Since datasets N5-N12 and B1-B2 have a large number of features (varying from 1,001 to 37187 - see Table 4.1), we use a simple and fast univariate filter approach to select a subset of the most relevant features before running our proposed feature selection methods.

The main objective of this initial univariate filter stage is to remove features which have a low correlation with class labels before running (any version of) the ML-CFS method. The average correlation between each feature and all labels is measured using Equation (4.8). The features are then ranked in decreasing order of average correlation with the class labels and then only the top k features are selected, where k is a user specified parameter, whose value defines the size of the feature space to be searched by the greedy search strategy implemented in the ML-CFS methods. We did experiments where the number of features selected by the univariate filter method, i.e; the feature space size varied between 100, 200, 300 and 400. It should be note that this kind of initial univariate filter stage is also often used in the conventional single-label classification literature [15, 108, 118]

4.4 Computational Results Comparing the First Version of ML-CFS and the Two Generic Extensions of ML-CFS

4.4.1 Experimental Methodology

The experiments reported in this Section 4.4 are devided into two parts, as follows. First, we ran an experiment for comparing the first version of ML-CFS (described in Section 4.1) with one extension of the ML-CFS method which uses the absolute value of correlation coefficient, which was described in Section 4.2.1. Second, we compare the best of those two ML-CFS versions, which turned out to be ML-CFS with absolute value of the correlation coefficient, againt two versions of ML-CFS based on Mutual Information, which were described in Section 4.2.2.

In each of these two types of experiments, in order to evaluate the predictive performance of the different versions of ML-CFS, the feature subset selected by each ML-CFS version was given to two different types of multi-label classification algorithm, namely the Multi-Label k-Nearest Neighbour (ML-kNN) classification algorithm proposed by [124] and the Back-Propagation Multi-Label Learning (BPMLL) Classification algorithm [123]. These two algorithms were run using their default parameters, which were mentioned in their corresponding paper. After that, the predictive accuracy of each classification model was measured, for each ML-CFS version, on the test set, containing data instances which were not included in the training set, therefore measuring the generalization ability of the classification model. For all datasets mentioned in Table 4.1, we used the predefined partition of each dataset into trainning and test sets provided by the MULAN repository website: `http://mulan.sourceforge.net/datasets.html`.

From a multi-label classification perspective we can measure the predictive ac-

curacy using different accuracy measures, such as: Hamming-loss, Ranking-loss, One-error, Coverage and Average Precision [113], as reviewed in Chapter 2. To evaluate the effectiveness of our proposed multi-label CFS method, we compare all mentioned predictive accuracy measure values obtained by ML-KNN and BPMLL when using each of the three above mentioned versions of the ML-CFS feature selection method. In all 14 datasets (N1-N12 and B1-B2) used in this experiment we used the zero-mean normalization method, described earlier.

4.4.2 Experimental Results for the First version of ML-CFS and ML-CFS with the Absolute Value of Correlation Coefficient using ML-kNN Classifier

Tables 4.2 - 4.6 show the predictive performance of the first version of ML-CFS and the ML-CFS with absolute value of correlation coefficient. Table 4.2 shows results for the four datasets having less than 300 features. In these datasets, ML-CFS was applied to the full set of features. However, the other datasets have more than 1,000 features. For these datasets with very large dimentionality, Tables 4.3 - 4.6 report results for different feature space sizes, i.e, different numbers of features pre-selected by the previously mentioned univariate filter method, namely 100, 200, 300 and 400 features, respectively. For each ML-CFS version, each table reports the values of each of the five measures of multi-label predictive accuracy mentioned earlier.

In Tables 4.2 - 4.6, ML-CFS stands for the first version of ML-CFS; and ML-CFSabs stands for ML-CFS with the Absolute value of correlation coefficient. The numbers in each column titled "R" denote the ranks achieved by each method according to the accuracy measure in the corresponding left column. The ranks vary in the range from 1 (best) to 2 (worst). The tables also report, in the last column, the average rank (AR) of each method across all five predictive accuracy

Table 4.2: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using ML-kNN as the classifier - small datasets (with less than 300 features)

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Cov.**	**R**	**H-Loss**	**R**	**OneErr.**	**R**	**R-Loss**	**R**	**AR**
CAL500	ML-CFS	0.485	2	131.368	2	0.140	2	0.124	2	0.1862	2	2
	ML-CFSabs	0.491	1	130.954	1	0.139	1	0.116	1	0.1845	1	1
Scene	ML-CFS	0.756	2	0.902	2	0.134	2	0.389	2	0.1595	2	2
	ML-CFSabs	0.798	1	0.731	1	0.117	1	0.329	1	0.1254	1	1
Emotions	ML-CFS	0.740	2	2.267	2	0.264	2	0.342	2	0.2341	2	2
	ML-CFSabs	0.796	1	1.975	1	0.210	1	0.272	1	0.1777	1	1
Yeast	ML-CFS	0.745	2	6.530	2	0.205	2	0.241	2	0.1811	2	2
	ML-CFSabs	0.758	1	6.459	1	0.200	1	0.230	1	0.1733	1	1
MEAN	ML-CFS		2		2		2		2		2	2
	ML-CFSabs		1		1		1		1		1	1

measures, for each dataset. The last two rows of each table show the mean rank for each method across all the datasets, In those last two rows, the mean value of each accuracy measure is not reported because that mean value would not be vary meaningful, since the different datasets have different degrees of difficult for a classification algorithm, so that different accuracies across datasets cannot be fairly compared. On the other hand, it is fair to compare the ML-CFS version across all datasets, so the mean ranks are reported. Finally the last column of the last two rows shows the average ranks over the five predictive accuracy measures and over all the datasets.

Clearly, in Tables 4.2 - 4.6, ML-CFSabs obtained substantially better predictive accuracy (substantially lower mean rank) than ML-CFS for each of the five accuracy measures in every table, as follows.

In Table 4.2 ML-CFSabs outperforms ML-CFS on all four datasets with overall average rank = 1.0. Also, ML-CFSabs obtains the better rank for all five predictive accuracy measures. In Table 4.3 - 4.6, when the feature space size was set to 100, 200, 300 and 400 respectively, ML-CFSabs obtained clearly better predictive accuracy (lower overall average rank) than ML-CFS for every feature space size. More precisely, ML-CFSabs outperforms ML-CFS on 8 - 9 datasets (out of 10

Table 4.3: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using ML-kNN as the classifier - feature space size = 100

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Cov.	R	H-Loss	R	OneErr.	R	R-Loss	R	AR
Business	ML-CFS	0.867	1	2.490	2	0.028	1	0.129	1	0.046	2	1.4
	ML-CFSabs	0.866	2	2.418	1	0.029	2	0.135	2	0.044	1	1.6
Art	ML-CFS	0.443	2	6.098	2	0.063	2	0.735	2	0.175	2	2
	ML- CFSabs	0.523	1	5.307	1	0.060	1	0.610	1	0.133	1	1
Education	ML-CFS	0.494	2	4.316	2	0.044	2	0.659	2	0.103	2	2
	ML- CFSabs	0.544	1	3.872	1	0.041	1	0.604	1	0.091	1	1
Recreation	ML-CFS	0.413	2	4.983	2	0.064	2	0.760	2	0.193	2	2
	ML- CFSabs	0.536	1	4.327	1	0.059	1	0.600	1	0.158	1	1
Health	ML-CFS	0.628	2	3.825	2	0.050	2	0.480	2	0.077	2	2
	ML- CFSabs	0.629	1	3.803	1	0.050	1	0.479	1	0.075	1	1
Enter.ment	ML-CFS	0.492	2	3.770	2	0.065	2	0.699	2	0.147	2	2
	ML- CFSabs	0.578	1	3.186	1	0.056	1	0.570	1	0.120	1	1
Computer	ML-CFS	0.607	2	4.590	2	0.044	2	0.473	2	0.099	2	2
	ML- CFSabs	0.633	1	4.200	1	0.040	1	0.452	1	0.089	1	1
Science	ML-CFS	0.406	2	7.440	1	0.036	2	0.746	2	0.150	1	1.6
	ML- CFSabs	0.419	1	7.462	2	0.036	1	0.718	1	0.151	2	1.4
Enron	ML-CFS	0.562	2	14.192	2	0.059	2	0.435	2	0.106	1	1.8
	ML- CFSabs	0.570	1	13.553	1	0.058	1	0.389	1	0.107	2	1.2
Medical	ML-CFS	0.561	2	4.712	2	0.026	2	0.583	2	0.087	2	2
	ML- CFSabs	0.767	1	3.202	1	0.018	1	0.304	1	0.052	1	1
MEAN	ML-CFS		1.9		1.9		1.9		1.9		1.8	1.88
	ML- CFSabs		1.1		1.1		1.1		1.1		1.2	1.12

Table 4.4: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using ML-kNN as the classifier - feature space size = 200

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Cov.	R	H-Loss	R	OneErr.	R	R-Loss	R	AR
Business	ML-CFS	0.867	2	2.356	1	0.029	1	0.134	1	0.043	2	1.4
	ML-CFSabs	0.868	1	2.363	2	0.029	2	0.136	2	0.043	1	1.6
Art	ML-CFS	0.456	2	5.867	2	0.063	2	0.714	2	0.168	2	2
	ML- CFSabs	0.522	1	5.395	1	0.060	1	0.604	1	0.150	1	1
Education	ML-CFS	0.495	2	4.282	2	0.044	2	0.657	2	0.102	2	2
	ML- CFSabs	0.551	1	3.838	1	0.041	1	0.592	1	0.090	1	1
Recreation	ML-CFS	0.413	2	5.013	2	0.063	2	0.761	2	0.193	2	2
	ML- CFSabs	0.572	1	4.120	1	0.055	1	0.545	1	0.148	1	1
Health	ML-CFS	0.628	2	3.825	2	0.050	2	0.480	2	0.077	2	2
	ML- CFSabs	0.675	1	3.444	1	0.045	1	0.415	1	0.065	1	1
Enter.ment	ML-CFS	0.498	2	3.697	2	0.064	2	0.690	2	0.143	2	2
	ML- CFSabs	0.602	1	3.122	1	0.054	1	0.530	1	0.115	1	1
Computer	ML-CFS	0.608	2	4.533	2	0.043	2	0.472	2	0.098	2	2
	ML- CFSabs	0.631	1	4.280	1	0.039	1	0.451	1	0.091	1	1
Science	ML-CFS	0.407	2	7.355	1	0.036	1	0.741	2	0.148	1	1.4
	ML- CFSabs	0.422	1	7.401	2	0.036	2	0.713	1	0.149	2	1.6
Enron	ML-CFS	0.570	2	14.320	2	0.059	2	0.394	2	0.106	2	2
	ML- CFSabs	0.587	1	13.382	1	0.058	1	0.375	1	0.098	1	1
Medical	ML-CFS	0.640	2	3.736	2	0.023	2	0.498	2	0.066	2	2
	ML- CFSabs	0.820	1	2.772	1	0.015	1	0.225	1	0.045	1	1
MEAN	ML-CFS		2		1.8		1.8		1.9		1.9	1.88
	ML- CFSabs		1		1.2		1.2		1.1		1.1	1.12

Table 4.5: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using ML-kNN as the classifier - feature space size = 300

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Cov.	R	H-Loss	R	OneErr.	R	R-Loss	R	AR
Business	ML-CFS	0.867	2	2.409	2	0.029	1.5	0.132	1	0.043	2	1.7
	ML-CFSabs	0.868	1	2.371	1	0.029	1.5	0.136	2	0.041	1	1.3
Art	ML-CFS	0.453	2	5.830	2	0.063	2	0.717	2	0.167	2	2
	ML- CFSabs	0.509	1	5.487	1	0.060	1	0.621	1	0.153	1	1
Education	ML-CFS	0.496	2	4.254	2	0.044	2	0.654	2	0.102	2	2
	ML- CFSabs	0.560	1	3.766	1	0.040	1	0.580	1	0.088	1	1
Recreation	ML-CFS	0.425	2	4.865	2	0.063	2	0.748	2	0.188	2	2
	ML- CFSabs	0.585	1	3.988	1	0.055	1	0.530	1	0.143	1	1
Health	ML-CFS	0.672	2	3.454	2	0.045	2	0.419	2	0.067	2	2
	ML- CFSabs	0.681	1	3.358	1	0.045	1	0.415	1	0.063	1	1
Enter.ment	ML-CFS	0.501	2	3.657	2	0.064	2	0.687	2	0.142	2	2
	ML- CFSabs	0.609	1	3.023	1	0.054	1	0.529	1	0.111	1	1
Computer	ML-CFS	0.619	2	4.454	2	0.042	2	0.454	2	0.095	2	2
	ML- CFSabs	0.641	1	4.187	1	0.039	1	0.437	1	0.088	1	1
Science	ML-CFS	0.408	2	7.351	1	0.036	2	0.741	2	0.148	1	1.6
	ML- CFSabs	0.422	1	7.410	2	0.036	1	0.715	1	0.149	2	1.4
Enron	ML-CFS	0.567	2	14.124	2	0.060	2	0.409	2	0.106	2	2
	ML- CFSabs	0.584	1	13.218	1	0.059	1	0.383	1	0.097	1	1
Medical	ML-CFS	0.631	2	3.777	2	0.024	2	0.513	2	0.067	2	2
	ML- CFSabs	0.811	1	2.845	1	0.017	1	0.239	1	0.045	1	1
MEAN	ML-CFS		2		1.9		1.95		1.9		1.9	1.93
	ML- CFSabs		1		1.1		1.05		1.1		1.1	1.07

datasets) with overall average rank equal to 1.12, 1.12, 1.07 and 1.13 in Tables 4.3 through 4.6, respectively; while the first version of ML-CFS has much larger (worse) average ranks (1.88, 1.88, 1.93 and 1.87, respectively).

Table 4.7 shows the summary of results reported in Tables 4.2 through 4.6, by reporting the average rank and the average number of features selected by ML-CFS and ML-CFSabs on all datasets. For CAL500, Scene, Yeast and Emotion datasets, where all features were available to ML-CFS and ML-CFSabs, ML-CFSabs obtains the best average rank (1.0); while ML-CFS obtains the worst rank (2.0). For the other large datasets the table reports average results over all those datasets for each feature space size used in our experiments (feature space size = 100, 200, 300 and 400). In those datasets, ML-CFSabs obtains the best ranks (1.12, 1.12, 1.07 and 1.13, respectively); while ML-CFS obtains much worse ranks (1.88, 1.88, 1.93 and 1.83, respectively). In terms of the number of selected features, ML-CFSabs selected the larger number of features in most cases, except in the Scene dataset

Table 4.6: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using ML-kNN as the classifier - feature space size = 400

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Cov.	R	H-Loss	R	OneErr.	R	R-Loss	R	AR
Business	ML-CFS	0.868	1	2.386	2	0.029	1	0.131	1	0.043	1.5	1.3
	ML-CFSabs	0.866	2	2.385	1	0.029	2	0.137	2	0.043	1.5	1.7
Art	ML-CFS	0.452	2	5.849	2	0.063	2	0.719	2	0.167	2	2
	ML-CFSabs	0.517	1	5.414	1	0.060	1	0.613	1	0.150	1	1
Education	ML-CFS	0.497	2	4.226	2	0.044	2	0.656	2	0.100	2	2
	ML-CFSabs	0.563	1	3.796	1	0.040	1	0.573	1	0.089	1	1
Recreation	ML-CFS	0.413	2	5.046	2	0.063	2	0.760	2	0.195	2	2
	ML-CFSabs	0.587	1	4.010	1	0.053	1	0.527	1	0.145	1	1
Health	ML-CFS	0.670	2	3.440	2	0.045	2	0.426	2	0.066	2	2
	ML-CFSabs	0.709	1	3.177	1	0.042	1	0.372	1	0.058	1	1
Enter.ment	ML-CFS	0.504	2	3.657	2	0.065	2	0.680	2	0.142	2	2
	ML-CFSabs	0.620	1	2.974	1	0.054	1	0.511	1	0.109	1	1
Computer	ML-CFS	0.617	2	4.455	2	0.042	2	0.458	2	0.095	2	2
	ML-CFSabs	0.642	1	4.190	1	0.038	1	0.434	1	0.088	1	1
Science	ML-CFS	0.410	2	7.229	1	0.036	1	0.742	2	0.147	1	1.4
	ML-CFSabs	0.421	1	7.409	2	0.036	2	0.713	1	0.149	2	1.6
Enron	ML-CFS	0.563	2	14.394	2	0.059	2	0.406	2	0.107	2	2
	ML-CFSabs	0.586	1	13.321	1	0.058	1	0.380	1	0.098	1	1
Medical	ML-CFS	0.622	2	4.061	2	0.024	2	0.501	2	0.072	2	2
	ML-CFSabs	0.811	1	2.876	1	0.017	1	0.240	1	0.046	1	1
MEAN	ML-CFS		1.9		1.9		1.8		1.9		1.85	1.87
	ML-CFSabs		1.1		1.1		1.2		1.1		1.15	1.13

(as shown in column titled "S.F").

Hence, it seems that one factor contributing to the worse predictive accuracy obtained by ML-CFS is that it tends to select substantially fewer features than ML-CFSabs; i.e, ML-CFS seems to remove more relevant features than ML-CFSabs. The number of features selected by ML-CFS is particularly low on the CAL500 and Emotions datasets, where it selected only 3 and 5 features, respectively.

4.4.3 Experimental Results for the First version of ML-CFS and ML-CFS with the Absolute Value of Correlation Coefficient using the BPMLL Classifier

This Section reports the results of experiments using the same experimental methodology used to produce the results reported in the previous section. The difference

Table 4.7: Summary of average ranking (AR) and the number of selected features (Sel.F) for ML-CFS and ML-CFSabs when using ML-kNN as the classifier

Datasets and feature space size	**ML-CFS**		**ML-CFSabs**	
	AR	**Sel.F**	**AR**	**Sel.F**
Emotion	2.00	3.00	1.00	10.00
CAL500	2.00	5.00	1.00	10.00
Scene	2.00	23.00	1.00	22.00
Yeast	2.00	19.00	1.00	23.00
100	1.88	18.60	1.12	31.70
200	1.88	30.60	1.12	49.20
300	1.93	40.20	1.07	60.50
400	1.83	47.70	1.13	73.20

is that this section reports results obtained with the BPMLL classifier, rather than the ML-kNN classifier used in the previous section.

In Tables 4.8 - 4.12, ML-CFSabs obtains the best predictive accuracy and outperforms the first version of ML-CFS in all cases. For example; in Table 4.8, reporting results for small datasets (with less than 300 features) ML-CFSabs obtains overall rank 1.0; which is much better than ML-CFS, which has overall rank 2.0.

In Table 4.9 - 4.12, when we set the feature space size equal to 100, 200, 300 and 400 respectively (after applying the previously described univariate filter method in a pre-processing step), ML-CFSabs obtained better predictive accuracy (lower average rank) than ML-CFS for every feature space size, i.e., in all four tables. In addition, ML-CFSabs outperforms the first version of ML-CFS for nearly all five predictive accuracy measures in Tables 4.9 - 4.12. The only exceptions are that ML-CFSabs and ML-CFS obtain the same mean rank (1.5) for the H-loss measure in Tables 4.9 and 4.10 as well as for the OneErr (OneError) measure in Table 4.10.

Table 4.13 shows the summary of results reported in Tables 4.9 through 4.12, showing the average rank and the average number of features selected by ML-CFS

Table 4.8: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using BPMLL as the classifier - small datasets

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
CAL500	ML-CFS	0.498	2.0	129.659	2.0	0.283	2.0	0.116	1.0	0.180	2.0	1.8
	ML-CFSabs	0.500	1.0	129.458	1.0	0.273	1.0	0.118	2.0	0.180	1.0	1.2
Scene	ML-CFS	0.745	2.0	0.873	2.0	0.169	2.0	0.421	2.0	0.153	2.0	2.0
	ML-CFSabs	0.771	1.0	0.761	1.0	0.154	1.0	0.389	1.0	0.132	1.0	1.0
Emotions	ML-CFS	0.761	2.0	2.104	2.0	0.272	2.0	0.337	2.0	0.212	2.0	2.0
	ML-CFSabs	0.776	1.0	2.005	1.0	0.225	1.0	0.328	1.0	0.189	1.0	1.0
Yeast	ML-CFS	0.740	2.0	6.630	1.0	0.231	2.0	0.245	2.0	0.187	2.0	1.8
	ML-CFSabs	0.742	1.0	6.643	2.0	0.230	1.0	0.244	1.0	0.183	1.0	1.2
MEAN	ML-CFS		2.0		1.8		2.0		1.8		2.0	1.9
	ML-CFSabs		1.0		1.3		1.0		1.3		1.0	1.1

and ML-CFSabs on all datasets when using BPMLL as the classifier. Clearly, ML-CFSabs outperformed ML-CFS with the best average ranking in every case. In terms of the number of selected features, ML-CFSabs selected a larger number of features in most cases, except on the Scene dataset (as shown in column titled "S.F"), as point out earlier in the discussion of Table 4.7 (with the summary of results for the ML-kNN classifier). It should be noted that the value of the "S.F". column in Table 4.13 (for BPMLL classifier) are exactly the same as the values of that column in Table 4.7, since both ML-CFS and ML-CFSabs are filter feature selection methods that select a set of features independent from the classifier that will use the selected features.

4.4.4 Experimental Results Comparing ML-CFS with the Absolute Value of Correlation Coefficient and ML-CFS Using Mutual Information for Class Label Weighting Using the ML-kNN Classifier

The previous two Sections (4.4.2 and 4.4.3) have reported results clearly showing that ML-CFS with Absolute Value of Correlation Coefficient obtained in general much better predictive accuracy than the first version of ML-CFS, both when using ML-kNN and when using BPMLL as the multi-label classifier. Hence, Tables 4.14 - 4.18 report results comparing ML-CFSabs and ML-CFS using MI (Mutual

Table 4.9: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using BPMLL as the classifier - feature space size = 100

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Cov.	R	H-Loss	R	OneErr.	R	R-Loss	R	AR
Business	ML-CFS	0.847	2	2.875	2	0.037	1	0.139	1.5	0.051	1.5	1.6
	ML-CFSabs	0.848	1	2.872	1	0.040	2	0.139	1.5	0.051	1.5	1.4
Art	ML-CFS	0.436	1	5.985	1	0.138	1	0.752	1.5	0.174	1	1.1
	ML- CFSabs	0.436	2	5.989	2	0.189	2	0.752	1.5	0.175	2	1.9
Education	ML-CFS	0.477	2	4.648	2	0.112	1	0.681	1.5	0.110	2	1.7
	ML- CFSabs	0.479	1	4.603	1	0.123	2	0.681	1.5	0.109	1	1.3
Recreation	ML-CFS	0.378	1	5.415	1	0.178	1	0.804	1	0.217	1	1
	ML- CFSabs	0.376	2	5.571	2	0.184	2	0.805	2	0.222	2	2
Health	ML-CFS	0.619	2	3.953	2	0.413	2	0.485	1	0.077	2	1.8
	ML- CFSabs	0.620	1	3.935	1	0.129	1	0.488	2	0.077	1	1.2
Enter.ment	ML-CFS	0.473	2	3.939	2	0.198	2	0.715	2	0.153	2	2
	ML- CFSabs	0.530	1	3.437	1	0.155	1	0.648	1	0.130	1	1
Computer	ML-CFS	0.598	2	4.893	1.5	0.091	2	0.475	1.5	0.102	1	1.6
	ML- CFSabs	0.599	1	4.893	1.5	0.073	1	0.475	1.5	0.103	2	1.4
Science	ML-CFS	0.398	1	7.688	1	0.119	1	0.758	1.5	0.155	1	1.1
	ML- CFSabs	0.397	2	7.747	2	0.124	2	0.758	1.5	0.156	2	1.9
Enron	ML-CFS	0.566	2	13.350	2	0.568	2	0.419	2	0.099	2	2
	ML- CFSabs	0.574	1	13.302	1	0.090	1	0.380	1	0.098	1	1
Medical	ML-CFS	0.544	2	4.134	2	0.052	2	0.637	2	0.075	2	2
	ML- CFSabs	0.733	1	2.642	1	0.025	1	0.384	1	0.042	1	1
MEAN	ML-CFS		1.7		1.65		1.5		1.55		1.55	1.59
	ML- CFSabs		1.3		1.35		1.5		1.45		1.45	1.41

Table 4.10: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using BPMLL as the classifier - feature space size = 200

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Cov.	R	H-Loss	R	OneErr.	R	R-Loss	R	AR
Business	ML-CFS	0.847	2	2.873	2	0.041	1	0.139	1.5	0.051	2	1.7
	ML-CFSabs	0.849	1	2.818	1	0.045	2	0.139	1.5	0.050	1	1.3
Art	ML-CFS	0.436	2	6.006	2	0.191	1	0.752	1.5	0.175	2	1.7
	ML- CFSabs	0.437	1	5.963	1	0.195	2	0.752	1.5	0.174	1	1.3
Education	ML-CFS	0.476	2	4.679	2	0.114	1	0.681	2	0.111	2	1.8
	ML- CFSabs	0.476	1	4.667	1	0.132	2	0.681	1	0.110	1	1.2
Recreation	ML-CFS	0.377	2	5.461	1	0.184	1	0.805	1	0.219	2	1.4
	ML- CFSabs	0.380	1	5.486	2	0.217	2	0.805	2	0.217	1	1.6
Health	ML-CFS	0.617	2	3.943	2	0.411	2	0.488	1	0.077	2	1.8
	ML- CFSabs	0.618	1	3.900	1	0.115	1	0.489	2	0.075	1	1.2
Enter.ment	ML-CFS	0.473	2	3.932	2	0.194	2	0.715	2	0.153	2	2
	ML- CFSabs	0.529	1	3.449	1	0.165	1	0.648	1	0.131	1	1
Computer	ML-CFS	0.598	1	4.949	1	0.077	1	0.475	1.5	0.103	1	1.1
	ML- CFSabs	0.595	2	5.003	2	0.080	2	0.475	1.5	0.106	2	1.9
Science	ML-CFS	0.396	1	7.803	1	0.131	2	0.758	1.5	0.156	1	1.3
	ML- CFSabs	0.396	2	7.866	2	0.126	1	0.758	1.5	0.158	2	1.7
Enron	ML-CFS	0.567	1	13.719	2	0.161	2	0.390	1	0.102	2	1.6
	ML- CFSabs	0.559	2	13.116	1	0.088	1	0.402	2	0.097	1	1.4
Medical	ML-CFS	0.550	2	3.808	2	0.041	2	0.655	2	0.069	2	2
	ML- CFSabs	0.748	1	2.650	1	0.024	1	0.359	1	0.043	1	1
MEAN	ML-CFS		1.7		1.7		1.5		1.5		1.8	1.64
	ML- CFSabs		1.3		1.3		1.5		1.5		1.2	1.36

Table 4.11: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using BPMLL as the classifier - feature space size = 300

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Cov.**	**R**	**H-Loss**	**R**	**OneErr.**	**R**	**R-Loss**	**R**	**AR**
Business	ML-CFS	0.848	2	2.846	2	0.042	2	0.139	1.5	0.050	2	1.9
	ML-CFSabs	0.849	1	2.825	1	0.038	1	0.139	1.5	0.050	1	1.1
Art	ML-CFS	0.436	1	6.009	2	0.192	1	0.752	1.5	0.176	2	1.5
	ML- CFSabs	0.436	2	5.963	1	0.207	2	0.752	1.5	0.174	1	1.5
Education	ML-CFS	0.476	2	4.695	2	0.129	2	0.681	1.5	0.111	2	1.9
	ML- CFSabs	0.481	1	4.560	1	0.122	1	0.681	1.5	0.108	1	1.1
Recreation	ML-CFS	0.376	2	5.571	1	0.180	1	0.805	1.5	0.222	1	1.3
	ML- CFSabs	0.376	1	5.662	2	0.269	2	0.805	1.5	0.225	2	1.7
Health	ML-CFS	0.618	2	3.967	2	0.127	2	0.489	1	0.077	2	1.8
	ML- CFSabs	0.623	1	3.908	1	0.126	1	0.489	2	0.074	1	1.2
Enter.ment	ML-CFS	0.472	2	3.953	2	0.209	2	0.715	2	0.154	2	2
	ML- CFSabs	0.518	1	3.559	1	0.188	1	0.662	1	0.136	1	1
Computer	ML-CFS	0.595	1	5.025	2	0.086	2	0.475	1.5	0.105	1	1.5
	ML- CFSabs	0.595	2	5.003	1	0.083	1	0.475	1.5	0.106	2	1.5
Science	ML-CFS	0.396	2	7.866	2	0.133	2	0.758	1.5	0.158	2	1.9
	ML- CFSabs	0.396	1	7.815	1	0.129	1	0.758	1.5	0.157	1	1.1
Enron	ML-CFS	0.556	2	13.792	2	0.089	1	0.406	2	0.102	2	1.8
	ML- CFSabs	0.568	1	13.231	1	0.089	2	0.392	1	0.098	1	1.2
Medical	ML-CFS	0.558	2	3.746	2	0.047	2	0.643	2	0.069	2	2
	ML- CFSabs	0.804	1	2.347	1	0.020	1	0.271	1	0.036	1	1
MEAN	ML-CFS		1.8		1.9		1.7		1.6		1.8	1.76
	ML- CFSabs		1.2		1.1		1.3		1.4		1.2	1.24

Table 4.12: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using BPMLL as the classifier - feature space size = 400

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Cov.**	**R**	**H-Loss**	**R**	**OneErr.**	**R**	**R-Loss**	**R**	**AR**
Business	ML-CFS	0.848	2	2.872	2	0.041	2	0.139	1.5	0.051	2	1.9
	ML-CFSabs	0.849	1	2.804	1	0.039	1	0.139	1.5	0.050	1	1.1
Art	ML-CFS	0.436	2	6.016	2	0.195	1	0.752	1.5	0.176	2	1.7
	ML- CFSabs	0.436	1	6.000	1	0.197	2	0.752	1.5	0.175	1	1.3
Education	ML-CFS	0.478	1	4.662	1	0.142	2	0.681	1.5	0.110	1	1.3
	ML- CFSabs	0.476	2	4.689	2	0.131	1	0.681	1.5	0.111	2	1.7
Recreation	ML-CFS	0.376	1	5.583	1	0.205	1	0.805	1.5	0.222	1	1.1
	ML- CFSabs	0.373	2	5.818	2	0.330	2	0.805	1.5	0.229	2	1.9
Health	ML-CFS	0.616	2	3.998	2	0.131	2	0.489	2	0.077	2	2
	ML- CFSabs	0.617	1	3.848	1	0.116	1	0.489	1	0.074	1	1
Enter.ment	ML-CFS	0.473	2	3.935	2	0.209	2	0.715	2	0.154	2	2
	ML- CFSabs	0.498	1	3.589	1	0.189	1	0.705	1	0.139	1	1
Computer	ML-CFS	0.595	2	5.011	2	0.070	1	0.475	1.5	0.106	2	1.7
	ML- CFSabs	0.596	1	4.980	1	0.086	2	0.475	1.5	0.106	1	1.3
Science	ML-CFS	0.396	1	7.810	2	0.136	2	0.758	1.5	0.157	1	1.5
	ML- CFSabs	0.396	2	7.787	1	0.129	1	0.758	1.5	0.157	2	1.5
Enron	ML-CFS	0.547	2	13.979	2	0.088	1	0.404	2	0.105	2	1.8
	ML- CFSabs	0.559	1	13.188	1	0.089	2	0.396	1	0.097	1	1.2
Medical	ML-CFS	0.566	2	3.941	2	0.041	2	0.630	2	0.070	2	2
	ML- CFSabs	0.795	1	2.504	1	0.019	1	0.276	1	0.040	1	1
MEAN	ML-CFS		1.7		1.8		1.6		1.7		1.7	1.7
	ML- CFSabs		1.3		1.2		1.4		1.3		1.3	1.3

Table 4.13: Summary of average ranking (AR) and the number of selected features (Sel.F) for ML-CFS and ML-CFSabs when using BPMLL as the classifier

Datasets and feature space size	**ML-CFS**		**ML-CFSabs**	
	AR	**Sel.F**	**AR**	**Sel.F**
CAL500	2.00	5.0	1.00	10.0
Scene	2.00	23.0	1.00	22.0
Emotions	2.00	3.0	1.00	10.0
Yeast	2.00	19.0	1.00	23.0
100	1.59	18.6	1.41	31.7
200	1.64	30.6	1.36	49.2
300	1.76	40.2	1.24	60.5
400	1.70	47.7	1.30	73.2

Information) for class label weighting. Recall that there are two versions of ML-CFS using MI. gmiML-CFS stands for the ML-CFS version where class labels with greater MI (Mutual Information) are assigned greater weights, while smiML-CFS stands for the ML-CFS version where class labels with smaller MI are assigned greater weights, as described in Section 4.2.2.

It is important to mention that gmiML-CFS and smiML-CFS also use the absolute value of the correlation coefficient (like ML-CFSabs). Hence, when comparing gmiML-CFS and smiML-CFS versus ML-CFSabs, we are evaluating the effectiveness of using mutual information for class label weighting in a controlled way.

The results are reported in Table 4.14 for the small datasets, where all features are available to the ML-CFS methods and in Table 4.15 through 4.18 for the large datasets (with more than 1000 features), where the univariate filter method was applied to reduce the feature space size, as described earlier.

The gmiML-CFS method obtains the best predictive accuracy and outperforms ML-CFSabs and smiML-CFS in general. For example; in Table 4.14 ML-CFS obtains overall average rank 1.60 (across all datasets and all accuracy measures), which is better than the ML-CFS and smiML-CFS methods, which obtain overall

Table 4.14: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - small datasets

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Emotion	gmiML-CFS	0.800	1.0	1.921	1.0	0.215	1.0	0.282	2.0	0.174	2.0	1.40
	smiML-CFS	0.789	2.0	1.965	2.0	0.228	2.0	0.277	1.0	0.173	1.0	1.60
	ML-CFSabs	0.740	3.0	2.267	3.0	0.264	3.0	0.342	3.0	0.234	3.0	3.00
CAL500	gmiML-CFS	0.492	2.0	131.068	2.0	0.139	1.0	0.120	2.0	0.185	2.0	1.80
	smiML-CFS	0.492	1.0	130.498	1.0	0.139	2.0	0.116	1.0	0.184	1.0	1.20
	ML-CFSabs	0.485	3.0	131.368	3.0	0.140	3.0	0.124	3.0	0.186	3.0	3.00
Scene	gmiML-CFS	0.499	2.0	2.161	2.0	0.210	2.0	0.715	2.0	0.411	2.0	2.00
	smiML-CFS	0.485	3.0	2.290	3.0	0.262	3.0	0.745	3.0	0.438	3.0	3.00
	ML-CFSabs	0.756	1.0	0.902	1.0	0.134	1.0	0.389	1.0	0.160	1.0	1.00
Yeast	gmiML-CFS	0.756	1.0	6.495	2.0	0.204	1.0	0.230	1.0	0.176	1.0	1.20
	smiML-CFS	0.747	2.0	6.470	1.0	0.210	3.0	0.253	3.0	0.179	2.0	2.20
	ML-CFSabs	0.745	3.0	6.530	3.0	0.205	2.0	0.241	2.0	0.181	3.0	2.60
MEAN	gmiML-CFS		1.50		1.75		1.25		1.75		1.75	1.60
	smiML-CFS		2.00		1.75		2.50		2.00		1.75	2.00
	ML-CFSabs		2.50		2.50		2.25		2.25		2.50	2.40

rank 2.4 and 2.0, respectively.

In Tables 4.15 - 4.18, when we set the feature space size equal to 100, 200, 300 and 400 respectively, gmiML-CFS obtained better predictive accuracy (lower overall average rank) than ML-CFSabs for every feature space size, i.e., in all four tables. In addition, gmiML-CFS outperforms the ML-CFSabs and smiML-CFS for all five predictive accuracy measures in Table 4.17, when we set feature space size equal to 300.

Table 4.19 reports the summary of results in terms of the overall average ranking and the number of selected features of ML-CFSabs and the two versions of ML-CFS using MI for class label weighting when using MLkNN as classifier. The table has one row for each of the small datasets, where all features were used as input. For the other (large) datasets, the table reports average results over all datasets for each feature space size used in the experiments.

Overall, in Table 4.19, gmiML-CFS obtained the best results, being the winner (with the smallest average rank) in 6 of 8 rows in that Table. For the small datasets, the difference between the average ranks of ML-CFSabs and gmiML-CFS

Table 4.15: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - feature space size = 100

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiML-CFS	0.583	1.0	13.679	2.0	0.057	1.0	0.389	1.5	0.100	1.0	1.30
	smiML-CFS	0.577	2.0	13.857	3.0	0.059	3.0	0.399	3.0	0.101	2.0	2.60
	ML-CFSabs	0.570	3.0	13.553	1.0	0.058	2.0	0.389	1.5	0.107	3.0	2.10
Medical	gmiML-CFS	0.760	2.0	3.372	2.0	0.017	1.0	0.301	1.0	0.055	2.0	1.60
	smiML-CFS	0.741	3.0	3.546	3.0	0.019	3.0	0.329	3.0	0.058	3.0	3.00
	ML-CFSabs	0.767	1.0	3.202	1.0	0.018	2.0	0.304	2.0	0.052	1.0	1.40
Business	gmiML-CFS	0.874	1.0	2.371	1.0	0.028	1.0	0.123	1.0	0.043	1.0	1.00
	smiML-CFS	0.872	2.0	2.391	2.0	0.028	2.0	0.126	2.0	0.043	2.0	2.00
	ML-CFSabs	0.866	3.0	2.418	3.0	0.029	3.0	0.135	3.0	0.044	3.0	3.00
Art	gmiML-CFS	0.528	1.0	5.398	2.0	0.059	1.0	0.588	1.0	0.150	2.0	1.40
	smiML-CFS	0.504	3.0	5.437	3.0	0.061	3.0	0.629	3.0	0.152	3.0	3.00
	ML-CFSabs	0.523	2.0	5.307	1.0	0.060	2.0	0.610	2.0	0.133	1.0	1.60
Education	gmiML-CFS	0.543	2.0	3.982	3.0	0.042	2.0	0.603	1.0	0.093	3.0	2.20
	smiML-CFS	0.541	3.0	3.905	2.0	0.042	3.0	0.606	3.0	0.092	2.0	2.60
	ML-CFSabs	0.544	1.0	3.872	1.0	0.041	1.0	0.604	2.0	0.091	1.0	1.20
Recreation	gmiML-CFS	0.535	2.0	4.349	3.0	0.059	1.0	0.601	2.0	0.159	3.0	2.20
	smiML-CFS	0.528	3.0	4.346	2.0	0.059	2.5	0.612	3.0	0.159	2.0	2.50
	ML-CFSabs	0.536	1.0	4.327	1.0	0.059	2.5	0.600	1.0	0.158	1.0	1.30
Health	gmiML-CFS	0.634	1.0	3.747	1.0	0.049	1.0	0.476	1.0	0.075	2.0	1.20
	smiML-CFS	0.628	3.0	3.811	3.0	0.050	3.0	0.480	3.0	0.077	3.0	3.00
	ML-CFSabs	0.629	2.0	3.803	2.0	0.050	2.0	0.479	2.0	0.075	1.0	1.80
Ent.ment	gmiML-CFS	0.593	1.0	3.158	1.0	0.056	1.0	0.548	1.0	0.119	1.0	1.00
	smiML-CFS	0.548	3.0	3.325	3.0	0.059	3.0	0.627	3.0	0.125	3.0	3.00
	ML-CFSabs	0.578	2.0	3.186	2.0	0.056	2.0	0.570	2.0	0.120	2.0	2.00
Computer	gmiML-CFS	0.623	3.0	4.416	2.0	0.040	2.0	0.450	2.0	0.094	2.0	2.20
	smiML-CFS	0.624	2.0	4.418	3.0	0.041	3.0	0.449	1.0	0.094	3.0	2.40
	ML-CFSabs	0.633	1.0	4.200	1.0	0.040	1.0	0.452	3.0	0.089	1.0	1.40
Science	gmiML-CFS	0.463	1.0	6.965	2.0	0.034	1.0	0.662	1.0	0.137	1.0	1.20
	smiML-CFS	0.443	2.0	6.952	1.0	0.035	2.0	0.700	2.0	0.137	2.0	1.80
	ML-CFSabs	0.419	3.0	7.462	3.0	0.036	3.0	0.718	3.0	0.151	3.0	3.00
MEAN	gmiML-CFS		1.50		1.90		1.20		1.25		1.80	1.53
	smiML-CFS		2.60		2.50		2.75		2.60		2.50	2.59
	ML-CFSabs		1.90		1.60		2.05		2.15		1.70	1.88

Table 4.16: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - feature space size = 200

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiML-CFS	0.580	3.0	13.719	3.0	0.059	3.0	0.396	2.0	0.101	3.0	2.80
	smiML-CFS	0.583	2.0	13.567	2.0	0.058	2.0	0.404	3.0	0.100	2.0	2.20
	ML-CFSabs	0.587	1.0	13.382	1.0	0.058	1.0	0.375	1.0	0.098	1.0	1.00
Medical	gmiML-CFS	0.807	2.0	2.929	3.0	0.016	3.0	0.231	2.0	0.046	2.0	2.40
	smiML-CFS	0.795	3.0	2.923	2.0	0.016	2.0	0.268	3.0	0.046	3.0	2.60
	ML-CFSabs	0.820	1.0	2.772	1.0	0.015	1.0	0.225	1.0	0.045	1.0	1.00
Business	gmiML-CFS	0.873	1.0	2.357	2.0	0.028	1.0	0.124	1.0	0.042	2.0	1.40
	smiML-CFS	0.871	2.0	2.326	1.0	0.028	2.0	0.132	2.0	0.041	1.0	1.60
	ML-CFSabs	0.868	3.0	2.363	3.0	0.029	3.0	0.136	3.0	0.043	3.0	3.00
Art	gmiML-CFS	0.537	1.0	5.323	2.0	0.059	1.0	0.579	1.0	0.148	2.0	1.40
	smiML-CFS	0.525	2.0	5.287	1.0	0.060	3.0	0.605	3.0	0.146	1.0	2.00
	ML-CFSabs	0.522	3.0	5.395	3.0	0.060	2.0	0.604	2.0	0.150	3.0	2.60
Education	gmiML-CFS	0.551	1.0	3.918	3.0	0.041	1.5	0.583	1.0	0.092	3.0	1.90
	smiML-CFS	0.544	3.0	3.899	2.0	0.042	3.0	0.600	3.0	0.092	2.0	2.60
	ML-CFSabs	0.551	2.0	3.838	1.0	0.041	1.5	0.592	2.0	0.090	1.0	1.50
Recreation	gmiML-CFS	0.572	2.0	4.223	3.0	0.054	1.0	0.540	1.0	0.152	3.0	2.00
	smiML-CFS	0.556	3.0	4.215	2.0	0.056	3.0	0.571	3.0	0.152	2.0	2.60
	ML-CFSabs	0.572	1.0	4.120	1.0	0.055	2.0	0.545	2.0	0.148	1.0	1.40
Health	gmiML-CFS	0.685	1.0	3.400	1.0	0.042	1.0	0.392	1.0	0.063	1.0	1.00
	smiML-CFS	0.672	3.0	3.469	3.0	0.045	3.0	0.413	2.0	0.066	3.0	2.80
	ML-CFSabs	0.675	2.0	3.444	2.0	0.045	2.0	0.415	3.0	0.065	2.0	2.20
Ent.ment	mi-ML-CFSabs	0.604	1.0	3.117	2.0	0.054	1.0	0.513	1.0	0.113	1.0	1.20
	smiML-CFS	0.583	3.0	3.096	1.0	0.058	3.0	0.580	3.0	0.115	2.5	2.50
	ML-CFSabs	0.602	2.0	3.122	3.0	0.054	2.0	0.530	2.0	0.115	2.5	2.30
Computer	gmiML-CFS	0.638	1.0	4.181	1.0	0.039	1.5	0.436	1.0	0.089	1.0	1.10
	smiML-CFS	0.630	3.0	4.258	2.0	0.040	3.0	0.446	2.0	0.091	3.0	2.60
	ML-CFSabs	0.631	2.0	4.280	3.0	0.039	1.5	0.451	3.0	0.091	2.0	2.30
Science	gmiML-CFS	0.484	1.0	6.808	2.0	0.034	1.5	0.638	1.0	0.133	2.0	1.50
	smiML-CFS	0.451	2.0	6.780	1.0	0.034	1.5	0.690	2.0	0.133	1.0	1.50
	ML-CFSabs	0.422	3.0	7.401	3.0	0.036	3.0	0.713	3.0	0.149	3.0	3.00
MEAN	gmiML-CFS		1.40		2.20		1.55		1.20		2.00	1.67
	smiML-CFS		2.60		1.70		2.55		2.60		2.05	2.30
	ML-CFSabs		2.00		2.10		1.90		2.20		1.95	2.03

Table 4.17: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - feature space size = 300

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiML-CFS	0.581	3.0	13.432	3.0	0.058	1.0	0.406	3.0	0.098	3.0	2.60
	smiML-CFS	0.587	1.0	13.290	2.0	0.059	3.0	0.387	2.0	0.097	1.5	1.90
	ML-CFSabs	0.584	2.0	13.218	1.0	0.059	2.0	0.383	1.0	0.097	1.5	1.50
Medical	gmiML-CFS	0.819	1.0	2.831	2.0	0.016	2.0	0.225	1.0	0.044	1.0	1.40
	smiML-CFS	0.814	2.0	2.769	1.0	0.015	1.0	0.234	2.0	0.044	2.0	1.60
	ML-CFSabs	0.811	3.0	2.845	3.0	0.017	3.0	0.239	3.0	0.045	3.0	3.00
Business	gmiML-CFS	0.876	1.0	2.292	1.0	0.028	1.0	0.127	1.0	0.040	1.0	1.00
	smiML-CFS	0.872	2.0	2.293	2.0	0.029	2.0	0.131	2.0	0.041	3.0	2.20
	ML-CFSabs	0.868	3.0	2.371	3.0	0.029	3.0	0.136	3.0	0.041	2.0	2.80
Art	gmiML-CFS	0.540	1.0	5.278	2.0	0.058	1.0	0.575	1.0	0.145	1.0	1.20
	smiML-CFS	0.525	2.0	5.246	1.0	0.060	2.0	0.603	2.0	0.145	2.0	1.80
	ML-CFSabs	0.509	3.0	5.487	3.0	0.060	3.0	0.621	3.0	0.153	3.0	3.00
Education	gmiML-CFS	0.552	2.0	3.895	3.0	0.041	2.0	0.588	2.0	0.091	3.0	2.40
	smiML-CFS	0.548	3.0	3.836	2.0	0.042	3.0	0.597	3.0	0.090	2.0	2.60
	ML-CFSabs	0.560	1.0	3.766	1.0	0.040	1.0	0.580	1.0	0.088	1.0	1.00
Recreation	gmiML-CFS	0.581	2.0	4.147	3.0	0.054	1.0	0.530	1.5	0.150	3.0	2.10
	smiML-CFS	0.575	3.0	4.122	2.0	0.055	2.5	0.543	3.0	0.149	2.0	2.50
	ML-CFSabs	0.585	1.0	3.988	1.0	0.055	2.5	0.530	1.5	0.143	1.0	1.40
Health	gmiML-CFS	0.699	1.0	3.303	1.0	0.042	1.0	0.380	1.0	0.061	1.0	1.00
	smiML-CFS	0.677	3.0	3.400	3.0	0.045	3.0	0.420	3.0	0.065	3.0	3.00
	ML-CFSabs	0.681	2.0	3.358	2.0	0.045	2.0	0.415	2.0	0.063	2.0	2.00
Ent.ment	gmiML-CFS	0.627	1.0	3.004	1.0	0.054	1.0	0.494	1.0	0.110	1.0	1.00
	smiML-CFS	0.587	3.0	3.062	3.0	0.058	3.0	0.574	3.0	0.113	3.0	3.00
	ML-CFSabs	0.609	2.0	3.023	2.0	0.054	2.0	0.529	2.0	0.111	2.0	2.00
Computer	gmiML-CFS	0.646	1.0	4.161	1.0	0.038	1.0	0.427	1.0	0.088	1.0	1.00
	smiML-CFS	0.630	3.0	4.284	3.0	0.040	3.0	0.450	3.0	0.091	3.0	3.00
	ML-CFSabs	0.641	2.0	4.187	2.0	0.039	2.0	0.437	2.0	0.088	2.0	2.00
Science	gmiML-CFS	0.489	1.0	6.622	1.0	0.034	1.5	0.629	1.0	0.129	1.0	1.10
	smiML-CFS	0.446	2.0	6.842	2.0	0.034	1.5	0.694	2.0	0.135	2.0	1.90
	ML-CFSabs	0.422	3.0	7.410	3.0	0.036	3.0	0.715	3.0	0.149	3.0	3.00
MEAN	gmiML-CFS		1.40		1.80		1.25		1.35		1.60	1.48
	smiML-CFS		2.40		2.10		2.40		2.50		2.35	2.35
	ML-CFSabs		2.20		2.10		2.35		2.15		2.05	2.17

Table 4.18: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - feature space size = 400

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiML-CFS	0.575	3.0	13.624	3.0	0.059	3.0	0.406	3.0	0.100	3.0	3.00
	smiML-CFS	0.586	2.0	13.263	1.0	0.058	2.0	0.396	2.0	0.098	2.0	1.80
	ML-CFSabs	0.586	1.0	13.321	2.0	0.058	1.0	0.380	1.0	0.098	1.0	1.20
Medical	gmiML-CFS	0.819	1.0	2.812	1.0	0.016	2.0	0.225	1.0	0.044	1.0	1.20
	smiML-CFS	0.811	2.0	2.871	2.0	0.016	1.0	0.240	2.5	0.046	2.0	1.90
	ML-CFSabs	0.811	3.0	2.876	3.0	0.017	3.0	0.240	2.5	0.046	3.0	2.90
Business	gmiML-CFS	0.877	1.0	2.299	2.0	0.028	1.0	0.123	1.0	0.041	2.0	1.40
	smiML-CFS	0.875	2.0	2.276	1.0	0.028	2.0	0.127	2.0	0.040	1.0	1.60
	ML-CFSabs	0.866	3.0	2.385	3.0	0.029	3.0	0.137	3.0	0.043	3.0	3.00
Art	gmiML-CFS	0.536	1.0	5.298	2.0	0.057	1.0	0.582	1.0	0.146	2.0	1.40
	smiML-CFS	0.527	2.0	5.213	1.0	0.059	2.0	0.602	2.0	0.143	1.0	1.60
	ML-CFSabs	0.517	3.0	5.414	3.0	0.060	3.0	0.613	3.0	0.150	3.0	3.00
Education	gmiML-CFS	0.559	2.0	3.848	3.0	0.040	2.0	0.574	2.0	0.090	3.0	2.40
	smiML-CFS	0.551	3.0	3.807	2.0	0.041	3.0	0.592	3.0	0.089	2.0	2.60
	ML-CFSabs	0.563	1.0	3.796	1.0	0.040	1.0	0.573	1.0	0.089	1.0	1.00
Recreation	gmiML-CFS	0.590	1.0	4.052	3.0	0.054	2.0	0.517	1.0	0.146	3.0	2.00
	smiML-CFS	0.570	3.0	3.971	1.0	0.055	3.0	0.552	3.0	0.144	1.0	2.20
	ML-CFSabs	0.587	2.0	4.010	2.0	0.053	1.0	0.527	2.0	0.145	2.0	1.80
Health	gmiML-CFS	0.721	1.0	3.193	2.0	0.040	1.0	0.348	1.0	0.058	1.0	1.20
	smiML-CFS	0.691	3.0	3.267	3.0	0.044	3.0	0.400	3.0	0.062	3.0	3.00
	ML-CFSabs	0.709	2.0	3.177	1.0	0.042	2.0	0.372	2.0	0.058	2.0	1.80
Ent.ment	gmiML-CFS	0.624	1.0	3.002	2.0	0.053	1.0	0.491	1.0	0.109	2.0	1.40
	smiML-CFS	0.585	3.0	3.060	3.0	0.057	3.0	0.578	3.0	0.113	3.0	3.00
	ML-CFSabs	0.620	2.0	2.974	1.0	0.054	2.0	0.511	2.0	0.109	1.0	1.60
Computer	gmiML-CFS	0.646	1.0	4.184	2.0	0.038	1.0	0.429	1.0	0.088	2.0	1.40
	smiML-CFS	0.641	3.0	4.134	1.0	0.040	3.0	0.437	3.0	0.087	1.0	2.20
	ML-CFSabs	0.642	2.0	4.190	3.0	0.038	2.0	0.434	2.0	0.088	3.0	2.40
Science	gmiML-CFS	0.485	1.0	6.741	1.0	0.034	1.5	0.629	1.0	0.132	1.0	1.10
	smiML-CFS	0.441	2.0	6.874	2.0	0.034	1.5	0.699	2.0	0.135	2.0	1.90
	ML-CFSabs	0.421	3.0	7.409	3.0	0.036	3.0	0.713	3.0	0.149	3.0	3.00
MEAN	gmiML-CFS		1.30		2.10		1.55		1.30		2.00	1.65
	smiML-CFS		2.50		1.70		2.35		2.55		1.80	2.18
	ML-CFSabs		2.20		2.20		2.10		2.15		2.20	2.17

Table 4.19: Summary of results in terms of average ranking (Avg.R) and the number of selected (S.F) features of ML-CFSabs and two versions of ML-CFS using Mutual Information for class label weighting using MLkNN as the classifier

Datasets and feature space size	gmi-ML-CFS		smiML-CFS		ML-CFSabs	
	S.F	Avg.R	S.F	Avg.R	S.F	Avg.R
Emotion	10.00	1.40	10.00	1.60	10.00	3.00
CAL500	12.90	1.80	12.40	1.20	10.00	3.00
Scene	36.00	2.00	24.00	3.00	22.00	1.00
Yeast	22.00	1.20	24.00	2.20	23.00	2.60
100	22.40	1.53	28.80	2.59	31.70	1.88
200	34.30	1.67	48.30	2.30	49.20	2.03
300	44.10	1.48	60.60	2.35	60.50	2.17
400	57.00	1.65	70.00	2.18	73.20	2.17

was quite large in Emotions, CAL500 and Yeast (1.60, 1.20 and 1.40, respectively). with gmiML-CFS winning (lower rank) in these 3 datasets. However, ML-CFSabs obtained the best average rank on the Scene dataset. In this dataset, the difference between the average ranks of ML-CFSabs and gmiML-CFS was 1.0. On the other hand, the difference between the average ranks of gmiML-CFS and smiML-CFS was small on the Emotions dataset (0.20), somewhat larger (0.6) on the CAL500 dataset, and larger on the Scene and Yeast datasets (1.0 on both datasets).

Turning to the last 4 rows of Table 4.19, the difference between the average ranks of ML-CFSabs and gmiML-CFS was small for the two smallest feature space sizes: a difference of 0.35 (1.88 - 1.53) for feature space size = 100 (Table 4.15) and a difference of 0.36 (2.03 - 1.67) for feature space size = 200 (Table 4.16). However, the difference between the average rank of ML-CFSabs and gmiML-CFS rise up to 0.69 (2.17 - 1.48) for feature space = 300 (Table 4.17) and a difference of 0.52 (2.17 - 1.65) for feature space size = 400 (Table 4.18). On the other hand, the difference between the average ranks of gmiML-CFS and smiML-CFS was large for all feature space sizes: the difference was 1.06, 0.63, 0.87 and 0.60 for feature space size equal to 100, 200, 300 and 400, respectively (Tables 4.15 - 4.18).

Table 4.20 presents a summary of the results from another perspective, reporting the average ranks (in terms of predictive accuracy) for each dataset. In each cell of the table, the first value is the average rank computed by averaging the corresponding ranks in Tables 4.15 - 4.18 (i.e, averaging over four feature space sizes); whilst the value between brackets is the "rank of the average ranks". This latter value was used for the statistical test of significance mentioned next.

Using the results shown in Table 4.20, we run the Friedman test and confidently conclude that there is a significant difference among the 3 methods on the 14 evaluation datasets at the 0.05 level of significance for a two tailed test (p value = 0.01817). Running the Holm's posthoc test on these data using gmiML-CFS as the

Table 4.20: Summary of overall average ranking (AR) across four feature space size for two versions of ML-CFS using MI for class label weighting and ML-CFSabs method using ML-kNN as the classifier

Dataset	**Overall Average Rank (AR) across 4 individual lengths**		
	gmiML-CFS	**smiML-CFS**	**ML-CFSabs**
CAL500	1.8(2)	1.2(1)	3(3)
Scene	2(2)	3(3)	1(1)
Emotions	1.4(1)	1.6(2)	3(3)
Yeast	1.2(1)	2.2(2)	2.6(3)
Enron	2.43(3)	2.13(2)	1.45(1)
Medical	1.65(1)	2.28(3)	2.08(2)
Business	1.2(1)	1.85(2)	2.95(3)
Art	1.35(1)	2.1(2)	2.55(3)
Education	2.23(2)	2.6(3)	1.18(1)
Recreation	2.08(2)	2.45(3)	1.48(1)
Health	1.1(1)	2.95(3)	1.95(2)
Ent.ment	1.15(1)	2.88(3)	1.98(2)
Computer	1.43(1)	2.55(3)	2.03(2)
Science	1.23(1)	1.78(2)	3(3)
Average	1.59(1.43)	2.25(2.43)	2.16(2.14)

control method, there are no significant differences when comparing gmiML-CFS versus ML-CFSabs at the 0.05 significance level, but there is a significant difference between gmiML-CFS versus smiML-CFS at the same level of significance (p value = 0.02445).

4.4.5 Experimental Results Comparing ML-CFS with the Absolute Value of Correlation Coefficient and ML-CFS Using Mutual Information for Class Label Weighting Using the BPMLL Classifier

This Section's contents is analogous to the contents of the previous Section 4.4.4. The difference is that this Section reports results using BPMLL classifier, rather than the ML-kNN classifier.

In Tables 4.21 - 4.25, the gmiML-CFS method obtains the best predictive accuracy and outperforms ML-CFSabs and smiML-CFS in most cases, although in

these tables the superiority of gmiML-CFS over the other ML-CFS methods is not as clear as when using ML-kNN in the previous section. In Table 4.21 (with results for the small datasets), ML-CFSabs obtains a somewhat better overall average rank (1.90) than the gmiML-CFS and smiML-CFS methods, which have overall average rank 2.0 and 2.2, respectively.

Tables 4.22 - 4.25 show the experimental results when we set the feature space size equal to 100, 200, 300 and 400, respectively; when using the univariate filter method to pre-process the large datasets. In these experiments, gmiML-CFS obtained better predictive accuracy (lower overall average rank) than ML-CFSabs for every feature space size, i.e., in all four tables; although the difference was small in two out of the four tables.

More precisely, the difference between the average ranks of ML-CFSabs and gmiML-CFS was small for the two smallest feature space sizes: a difference of 0.12 (2.02 - 1.90) for feature space size = 100 (Table 4.22) and a difference of 0.13 (2.14 - 2.01) for feature space size = 200 (Table 4.23). However, the difference between the average rank of ML-CFSabs and gmiML-CFS was substantially larger for the two largest feature space sizes. More precisely, the difference was 0.52 (2.24 - 1.72) for feature space = 300 (Table 4.24) and a difference of 0.66 (2.20 - 1.54) for feature space size = 400 (Table 4.25).

Note also that gmiML-CFS obtains the best rank for all five predictive accuracy measures in Tables 4.24 - 4.25. However, when the feature space size equals to 200 smiML-CFS obtains the best (smallest) overall average rank (1.85), while gmiML-CFS and ML-CFSabs obtain overall average rank 2.01 and 2.14, respectively.

Table 4.26 reports the summary of results in Table 4.21 - 4.25. Table 4.26 shows a similar pattern to Table 4.19 (for MLkNN). That is, focusing on the results in

Table 4.21: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using BPMLL as the classifier - small datasets

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Emotion	gmiML-CFS	0.795	1.0	1.930	1.0	0.220	1.0	0.294	1.0	0.172	1.0	1.00
	smiML-CFS	0.788	2.0	1.985	2.0	0.225	2.0	0.313	2.0	0.185	2.0	2.00
	ML-CFSabs	0.776	3.0	2.005	3.0	0.225	3.0	0.328	3.0	0.189	3.0	3.00
CAL500	gmiML-CFS	0.501	2.0	129.622	2.0	0.280	3.0	0.122	3.0	0.179	1.0	2.20
	smiML-CFS	0.501	1.0	129.631	3.0	0.278	2.0	0.117	1.0	0.179	2.0	1.80
	ML-CFSabs	0.500	3.0	129.458	1.0	0.273	1.0	0.118	2.0	0.180	3.0	2.00
Scene	gmiML-CFS	0.497	3.0	1.918	3.0	0.272	3.0	0.789	3.0	0.361	3.0	3.00
	smiML-CFS	0.517	2.0	1.725	2.0	0.270	2.0	0.771	2.0	0.326	2.0	2.00
	ML-CFSabs	0.771	1.0	0.761	1.0	0.154	1.0	0.389	1.0	0.132	1.0	1.00
Yeast	gmiML-CFS	0.742	2.0	6.644	3.0	0.228	1.0	0.245	2.0	0.183	3.0	2.20
	smiML-CFS	0.740	3.0	6.596	1.0	0.232	3.0	0.246	3.0	0.183	1.0	2.20
	ML-CFSabs	0.742	1.0	6.643	2.0	0.230	2.0	0.244	1.0	0.183	2.0	1.60
MEAN	gmiML-CFS		2.00		2.25		2.00		2.25		2.00	2.10
	smiML-CFS		2.00		2.00		2.25		2.00		1.75	2.00
	ML-CFSabs		2.00		1.75		1.75		1.75		2.25	1.90

the last four rows of Table 4.26, we can observe that the difference of average rank between gmiML-CFS and ML-CFSabs is small for the two smallest feature space size (100 and 200), but it is substantial for the two largest feature space sizes (300 and 400).

Table 4.27 shows the overall average rank of three versions of ML-CFS methods for each dataset average over all four feature space size - except for the first four (small) datasets, where all features were used as input. The first value in each cell is the actual average rank, whilst the value between brackets is the "rank of the average rank". This later value was used in the Friedman test. We conclude that there are no significant difference among the 3 algorithms on the 14 evaluation datasets at the 0.05 significance level for a two tailed test.

Table 4.22: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - feature space size = 100

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiML-CFS	0.567	3.0	13.207	1.0	0.089	2.0	0.403	3.0	0.097	1.0	2.00
	smiML-CFS	0.576	1.0	13.270	2.0	0.086	1.0	0.377	1.0	0.098	3.0	1.60
	ML-CFSabs	0.574	2.0	13.302	3.0	0.090	3.0	0.380	2.0	0.098	2.0	2.40
Medical	gmiML-CFS	0.557	3.0	3.604	3.0	0.050	3.0	0.655	3.0	0.066	3.0	3.00
	smiML-CFS	0.727	2.0	2.687	2.0	0.027	2.0	0.383	1.0	0.045	2.0	1.80
	ML-CFSabs	0.733	1.0	2.642	1.0	0.025	1.0	0.384	2.0	0.042	1.0	1.20
Business	gmiML-CFS	0.853	1.0	2.751	2.0	0.042	2.0	0.139	1.0	0.048	1.0	1.40
	smiML-CFS	0.852	2.0	2.731	1.0	0.044	3.0	0.139	2.5	0.049	2.0	2.10
	ML-CFSabs	0.848	3.0	2.872	3.0	0.040	1.0	0.139	2.5	0.051	3.0	2.50
Art	gmiML-CFS	0.436	1.0	6.006	2.0	0.184	1.0	0.752	1.5	0.175	2.0	1.50
	smiML-CFS	0.436	2.0	6.010	3.0	0.185	2.0	0.753	3.0	0.175	3.0	2.60
	ML-CFSabs	0.436	3.0	5.989	1.0	0.189	3.0	0.752	1.5	0.175	1.0	1.90
Education	gmiML-CFS	0.480	1.0	4.532	1.0	0.134	3.0	0.679	1.0	0.107	1.0	1.40
	smiML-CFS	0.478	3.0	4.622	3.0	0.119	1.0	0.681	2.0	0.110	3.0	2.40
	ML-CFSabs	0.479	2.0	4.603	2.0	0.123	2.0	0.681	3.0	0.109	2.0	2.20
Recreation	gmiML-CFS	0.380	2.0	5.402	2.0	0.190	3.0	0.802	2.0	0.215	2.0	2.20
	smiML-CFS	0.388	1.0	5.306	1.0	0.190	2.0	0.797	1.0	0.211	1.0	1.20
	ML-CFSabs	0.376	3.0	5.571	3.0	0.184	1.0	0.805	3.0	0.222	3.0	2.60
Health	gmiML-CFS	0.623	1.0	3.927	1.0	0.108	2.0	0.481	1.5	0.076	1.0	1.30
	smiML-CFS	0.621	2.0	3.963	3.0	0.099	1.0	0.481	1.5	0.077	3.0	2.10
	ML-CFSabs	0.620	3.0	3.935	2.0	0.129	3.0	0.488	3.0	0.077	2.0	2.60
Ent.ment	gmiML-CFS	0.529	3.0	3.460	3.0	0.149	1.0	0.649	3.0	0.132	3.0	2.60
	smiML-CFS	0.530	2.0	3.437	1.0	0.164	3.0	0.648	1.0	0.131	2.0	1.80
	ML-CFSabs	0.530	1.0	3.437	2.0	0.155	2.0	0.648	2.0	0.130	1.0	1.60
Computer	gmiML-CFS	0.599	1.0	4.867	1.0	0.084	3.0	0.475	2.0	0.101	1.0	1.60
	smiML-CFS	0.598	3.0	4.954	3.0	0.072	1.0	0.475	2.0	0.103	3.0	2.40
	ML-CFSabs	0.599	2.0	4.893	2.0	0.073	2.0	0.475	2.0	0.103	2.0	2.00
Science	gmiML-CFS	0.396	2.0	7.819	2.0	0.128	2.0	0.758	2.0	0.157	2.0	2.00
	smiML-CFS	0.396	3.0	7.857	3.0	0.129	3.0	0.758	2.0	0.157	3.0	2.80
	ML-CFSabs	0.397	1.0	7.747	1.0	0.124	1.0	0.758	2.0	0.156	1.0	1.20
MEAN	gmiML-CFS		1.80		1.80		2.20		2.00		1.70	1.90
	smiML-CFS		2.10		2.20		1.90		1.70		2.50	2.08
	ML-CFSabs		2.10		2.00		1.90		2.30		1.80	2.02

Table 4.23: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - feature space size = 200

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiML-CFS	0.559	3.0	13.293	3.0	0.087	1.0	0.405	3.0	0.098	3.0	2.60
	smiML-CFS	0.560	1.0	13.231	2.0	0.089	3.0	0.400	1.0	0.098	2.0	1.80
	ML-CFSabs	0.559	2.0	13.116	1.0	0.088	2.0	0.402	2.0	0.097	1.0	1.60
Medical	gmiML-CFS	0.758	2.0	2.475	1.0	0.026	3.0	0.341	2.0	0.040	1.0	1.80
	smiML-CFS	0.794	1.0	2.662	3.0	0.020	1.0	0.280	1.0	0.043	2.0	1.60
	ML-CFSabs	0.748	3.0	2.650	2.0	0.024	2.0	0.359	3.0	0.043	3.0	2.60
Business	gmiML-CFS	0.853	1.0	2.751	1.0	0.041	1.0	0.139	1.0	0.049	1.0	1.00
	smiML-CFS	0.848	3.0	2.858	3.0	0.042	2.0	0.139	2.0	0.051	3.0	2.60
	ML-CFSabs	0.849	2.0	2.818	2.0	0.045	3.0	0.139	3.0	0.050	2.0	2.40
Art	gmiML-CFS	0.437	3.0	5.985	3.0	0.185	1.0	0.752	2.0	0.175	3.0	2.40
	smiML-CFS	0.438	1.0	5.940	1.0	0.197	3.0	0.752	2.0	0.174	1.0	1.60
	ML-CFSabs	0.437	2.0	5.963	2.0	0.195	2.0	0.752	2.0	0.174	2.0	2.00
Education	gmiML-CFS	0.480	1.0	4.477	1.0	0.142	3.0	0.681	2.5	0.107	1.0	1.70
	smiML-CFS	0.478	2.0	4.625	2.0	0.132	1.0	0.681	2.5	0.109	2.0	1.90
	ML-CFSabs	0.476	3.0	4.667	3.0	0.132	2.0	0.681	1.0	0.110	3.0	2.40
Recreation	gmiML-CFS	0.379	3.0	5.530	3.0	0.206	1.0	0.803	1.0	0.219	3.0	2.20
	smiML-CFS	0.380	2.0	5.495	2.0	0.210	2.0	0.804	2.0	0.218	2.0	2.00
	ML-CFSabs	0.380	1.0	5.486	1.0	0.217	3.0	0.805	3.0	0.217	1.0	1.80
Health	gmiML-CFS	0.617	3.0	3.976	3.0	0.113	1.0	0.489	3.0	0.077	3.0	2.60
	smiML-CFS	0.618	1.0	3.890	1.0	0.118	3.0	0.488	1.0	0.075	1.0	1.40
	ML-CFSabs	0.618	2.0	3.900	2.0	0.115	2.0	0.489	2.0	0.075	2.0	2.00
Ent.ment	gmiML-CFS	0.506	3.0	3.533	3.0	0.172	2.0	0.688	3.0	0.135	3.0	2.80
	smiML-CFS	0.529	1.0	3.432	1.0	0.176	3.0	0.648	1.0	0.131	1.0	1.40
	ML-CFSabs	0.529	2.0	3.449	2.0	0.165	1.0	0.648	2.0	0.131	2.0	1.80
Computer	gmiML-CFS	0.601	1.0	4.810	1.0	0.084	3.0	0.475	2.0	0.101	1.0	1.60
	smiML-CFS	0.598	2.0	4.854	2.0	0.083	2.0	0.475	2.0	0.102	2.0	2.00
	ML-CFSabs	0.595	3.0	5.003	3.0	0.080	1.0	0.475	2.0	0.106	3.0	2.40
Science	gmiML-CFS	0.396	1.0	7.811	1.0	0.129	2.0	0.758	2.0	0.157	1.0	1.40
	smiML-CFS	0.396	2.0	7.814	2.0	0.134	3.0	0.758	2.0	0.157	2.0	2.20
	ML-CFSabs	0.396	3.0	7.866	3.0	0.126	1.0	0.758	2.0	0.158	3.0	2.40
MEAN	gmiML-CFS		2.10		2.00		1.80		2.15		2.00	2.01
	smiML-CFS		1.60		1.90		2.30		1.65		1.80	1.85
	ML-CFSabs		2.30		2.10		1.90		2.20		2.20	2.14

Table 4.24: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - feature space size = 300

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiML-CFS	0.567	2.0	13.217	2.0	0.089	1.0	0.400	2.0	0.097	2.0	1.80
	smiML-CFS	0.563	3.0	13.103	1.0	0.090	3.0	0.401	3.0	0.097	1.0	2.20
	ML-CFSabs	0.568	1.0	13.231	3.0	0.089	2.0	0.392	1.0	0.098	3.0	2.00
Medical	gmiML-CFS	0.796	3.0	2.344	1.0	0.021	3.0	0.282	3.0	0.036	2.0	2.40
	smiML-CFS	0.799	2.0	2.595	3.0	0.020	1.0	0.278	2.0	0.041	3.0	2.20
	ML-CFSabs	0.804	1.0	2.347	2.0	0.020	2.0	0.271	1.0	0.036	1.0	1.40
Business	gmiML-CFS	0.853	1.0	2.762	1.0	0.034	1.0	0.139	2.0	0.049	1.0	1.20
	smiML-CFS	0.850	2.0	2.789	2.0	0.042	3.0	0.139	2.0	0.050	2.0	2.20
	ML-CFSabs	0.849	3.0	2.825	3.0	0.038	2.0	0.139	2.0	0.050	3.0	2.60
Art	gmiML-CFS	0.433	3.0	6.014	3.0	0.205	2.0	0.752	2.0	0.177	3.0	2.60
	smiML-CFS	0.437	1.0	5.973	2.0	0.199	1.0	0.752	2.0	0.174	2.0	1.60
	ML-CFSabs	0.436	2.0	5.963	1.0	0.207	3.0	0.752	2.0	0.174	1.0	1.80
Education	gmiML-CFS	0.482	1.0	4.474	1.0	0.122	1.0	0.678	1.0	0.106	1.0	1.00
	smiML-CFS	0.476	3.0	4.702	3.0	0.131	3.0	0.681	2.5	0.111	3.0	2.90
	ML-CFSabs	0.481	2.0	4.560	2.0	0.122	2.0	0.681	2.5	0.108	2.0	2.10
Recreation	gmiML-CFS	0.379	1.0	5.561	2.0	0.222	1.0	0.802	1.0	0.220	2.0	1.40
	smiML-CFS	0.378	2.0	5.528	1.0	0.260	2.0	0.805	2.5	0.219	1.0	1.70
	ML-CFSabs	0.376	3.0	5.662	3.0	0.269	3.0	0.805	2.5	0.225	3.0	2.90
Health	gmiML-CFS	0.612	3.0	3.906	2.0	0.122	1.0	0.490	3.0	0.076	3.0	2.40
	smiML-CFS	0.624	1.0	3.898	1.0	0.129	3.0	0.489	1.0	0.074	1.0	1.40
	ML-CFSabs	0.623	2.0	3.908	3.0	0.126	2.0	0.489	2.0	0.074	2.0	2.20
Ent.ment	gmiML-CFS	0.529	2.0	3.455	2.0	0.154	1.0	0.649	2.0	0.132	2.0	1.80
	smiML-CFS	0.530	1.0	3.418	1.0	0.181	2.0	0.648	1.0	0.131	1.0	1.20
	ML-CFSabs	0.518	3.0	3.559	3.0	0.188	3.0	0.662	3.0	0.136	3.0	3.00
Computer	gmiML-CFS	0.600	1.0	4.861	1.0	0.083	1.0	0.475	2.0	0.102	1.0	1.20
	smiML-CFS	0.598	2.0	4.947	2.0	0.085	3.0	0.475	2.0	0.104	2.0	2.20
	ML-CFSabs	0.595	3.0	5.003	3.0	0.083	2.0	0.475	2.0	0.106	3.0	2.60
Science	gmiML-CFS	0.397	1.0	7.749	1.0	0.134	2.0	0.758	2.0	0.156	1.0	1.40
	smiML-CFS	0.395	3.0	7.835	3.0	0.139	3.0	0.758	2.0	0.159	3.0	2.80
	ML-CFSabs	0.396	2.0	7.815	2.0	0.129	1.0	0.758	2.0	0.157	2.0	1.80
MEAN	gmiML-CFS		1.80		1.60		1.40		2.00		1.80	1.72
	smiML-CFS		2.00		1.90		2.40		2.00		1.90	2.04
	ML-CFSabs		2.20		2.50		2.20		2.00		2.30	2.24

Table 4.25: Values of five multi-label predictive accuracy measures for ML-CFSabs and two versions of ML-CFS using mutual information for class label weighting using ML-kNN as the classifier - feature space size = 400

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiML-CFS	0.565	1.0	13.340	3.0	0.087	1.0	0.394	1.0	0.098	2.0	1.60
	smiML-CFS	0.556	3.0	13.295	2.0	0.091	3.0	0.418	3.0	0.098	3.0	2.80
	ML-CFSabs	0.559	2.0	13.188	1.0	0.089	2.0	0.396	2.0	0.097	1.0	1.60
Medical	gmiML-CFS	0.804	1.0	2.395	1.0	0.020	3.0	0.267	1.0	0.037	1.0	1.40
	smiML-CFS	0.799	2.0	2.569	3.0	0.018	1.0	0.276	3.0	0.041	3.0	2.40
	ML-CFSabs	0.795	3.0	2.504	2.0	0.019	2.0	0.276	2.0	0.040	2.0	2.20
Business	gmiML-CFS	0.857	1.0	2.668	1.0	0.037	1.0	0.139	2.0	0.047	1.0	1.20
	smiML-CFS	0.853	2.0	2.735	2.0	0.038	2.0	0.139	2.0	0.049	2.0	2.00
	ML-CFSabs	0.849	3.0	2.804	3.0	0.039	3.0	0.139	2.0	0.050	3.0	2.80
Art	gmiML-CFS	0.435	3.0	5.997	1.0	0.202	2.0	0.752	2.0	0.176	3.0	2.20
	smiML-CFS	0.436	2.0	6.008	3.0	0.210	3.0	0.752	2.0	0.175	2.0	2.40
	ML-CFSabs	0.436	1.0	6.000	2.0	0.197	1.0	0.752	2.0	0.175	1.0	1.40
Education	gmiML-CFS	0.482	1.0	4.531	1.0	0.111	1.0	0.681	2.0	0.107	1.0	1.20
	smiML-CFS	0.477	2.0	4.668	2.0	0.131	3.0	0.681	2.0	0.110	2.0	2.20
	ML-CFSabs	0.476	3.0	4.689	3.0	0.131	2.0	0.681	2.0	0.111	3.0	2.60
Recreation	gmiML-CFS	0.376	1.0	5.650	1.0	0.266	1.0	0.805	2.0	0.223	1.0	1.20
	smiML-CFS	0.375	2.0	5.743	2.0	0.293	2.0	0.805	2.0	0.227	2.0	2.00
	ML-CFSabs	0.373	3.0	5.818	3.0	0.330	3.0	0.805	2.0	0.229	3.0	2.80
Health	gmiML-CFS	0.621	1.0	3.821	1.0	0.120	2.0	0.487	1.0	0.073	1.0	1.20
	smiML-CFS	0.613	3.0	3.961	3.0	0.128	3.0	0.489	3.0	0.077	3.0	3.00
	ML-CFSabs	0.617	2.0	3.848	2.0	0.116	1.0	0.489	2.0	0.074	2.0	1.80
Ent.ment	gmiML-CFS	0.505	2.0	3.590	3.0	0.194	3.0	0.688	2.0	0.139	3.0	2.60
	smiML-CFS	0.529	1.0	3.436	1.0	0.182	1.0	0.648	1.0	0.131	1.0	1.00
	ML-CFSabs	0.498	3.0	3.589	2.0	0.189	2.0	0.705	3.0	0.139	2.0	2.40
Computer	gmiML-CFS	0.599	2.0	4.888	1.0	0.084	1.0	0.475	2.0	0.103	1.0	1.40
	smiML-CFS	0.599	1.0	4.898	2.0	0.088	3.0	0.475	2.0	0.103	2.0	2.00
	ML-CFSabs	0.596	3.0	4.980	3.0	0.086	2.0	0.475	2.0	0.106	3.0	2.60
Science	gmiML-CFS	0.397	1.0	7.733	1.0	0.146	2.0	0.758	2.0	0.156	1.0	1.40
	smiML-CFS	0.395	3.0	7.900	3.0	0.150	3.0	0.758	2.0	0.159	3.0	2.80
	ML-CFSabs	0.396	2.0	7.787	2.0	0.129	1.0	0.758	2.0	0.157	2.0	1.80
MEAN	gmiML-CFS		1.40		1.40		1.70		1.70		1.50	1.54
	smiML-CFS		2.10		2.30		2.40		2.20		2.30	2.26
	ML-CFSabs		2.50		2.30		1.90		2.10		2.20	2.20

Table 4.26: Summary of results in terms of average ranking (Avg.R)and the number of selected features (S.F) of ML-CFSabs and two versions of ML-CFS using Mutual Information for class label weighting using BPMLL as classifier

Datasets and feature space size	gmi-ML-CFS		smiML-CFS		ML-CFSabs	
	S.F	Avg.R	S.F	Avg.R	S.F	Avg.R
Emotion	10.00	1.00	10.00	2.00	10.00	3.00
CAL500	12.90	2.20	12.40	1.80	10.00	2.00
Scene	36.00	3.00	24.00	2.00	22.00	1.00
Yeast	22.00	2.20	24.00	2.20	23.00	1.60
100	22.40	1.90	28.80	2.08	31.70	2.02
200	34.30	2.01	48.30	1.85	49.20	2.14
300	44.10	1.72	60.60	2.04	60.50	2.24
400	57.00	1.54	70.00	2.26	73.20	2.20

Table 4.27: Summary of overall average ranking (Avg.R) across four individual lengths for two versions of ML-CFS using MI for class label weighting and ML-CFSabs methods using BPMLL as classifier

Dataset	**Overall Average Rank (AR) across 4 individual lengths**		
	gmiML-CFS	**smiML-CFS**	**ML-CFSabs**
CAL500	2.2(3)	1.8(1)	2(2)
CAL500	2.2(3)	1.8(1)	2(2)
Scene	3(3)	2(2)	1(1)
Emotions	1(1)	2(2)	3(3)
Yeast	2.2(2)	2.2(2)	1.6(1)
Enron	2(2)	2.1(3)	1.9(1)
Medical	2.15(3)	2(2)	1.85(1)
Business	1.2(1)	2.23(2)	2.58(3)
Art	2.18(3)	2.05(2)	1.78(1)
Education	1.33(1)	2.35(3)	2.33(2)
Recreation	1.75(2)	1.73(1)	2.53(3)
Health	1.88(1)	1.98(2)	2.15(3)
Ent.ment	2.45(3)	1.35(1)	2.2(2)
Computer	1.45(1)	2.15(2)	2.4(3)
Science	1.55(1)	2.65(3)	1.8(2)
Average	1.88(1.93)	2.04(2)	2.08(2)

4.5 Computational Results Comparing the Best Version of ML-CFS (gmiML-CFS) and Other Multi-Label Feature Selection Methods

4.5.1 Methods Being Compared and Experimental Methodology

In this Section we compare the best version of our ML-CFS method according to the results reported in previous Section, namely gmiML-CFS, with several other multi-label feature selection methods, namely Relief for Multi-Label feature selection (RFML) and three different baseline approaches: Binary Relevance (BR), Correlation-Based Feature Selection with the union operator (CFS-U) and No feature selection (NoFS). The details of each method or approach are described next.

The RFML method is a well-known multi-label feature selection method pro-

posed in [105]. This method was discussed in Section 3.4. We used the RFML implementation kindly provided by the authors; with its default parameter setting. After running RFML and obtaining the corresponsding feature ranking, we selected the top k features in the ranking, where k is the same number of features selected by gmiML-CFS.

Binary Relevance (BR) is provided in the multi-label classification repository. This approach was discussed in Section 3.2 and it essentially consists of the base classifier (in our case kNN and multi-layer perceptron, which are provided on the Weka website). The base classifier was used with its default parameter setting.

The CFS-U approach, which was first introduced by the author of this thesis in [57], consists of running a conventional single-label CFS method for selecting a feature subset for each class label separately and then returning the union of those selected feature subsets as the set of features to be given to the multi-label classification algorithm. The CFS implementation used in our experiments was the single-label CFSSubsetEval method in the well-known Weka data mining tool [44]. This method was used with its default parameters, and it evaluates candidate feature subsets according to Equation (4.1).

In the NoFS approach, we give all original features in the dataset to the multi-label classifier, in the case of the "small" (with less than 300 features) datasets (CAL500, Emotion, Scene and Yeast datasets); while in the case of all the large datasets (with more than 1,000 features), we apply the initial univariate approach, based on Equation (4.3), in order to select a subset of features to be given to the multi-label classifier; as explained earlier.

Hence, note that the name "NoFS" refers to the lack of use of a sophisticated and multivariate feature selection method like ML-CFS; it does not refer to a complete lack of feature selection in the case of the large datasets.

Similarly, to the previous Sections of results in this Chapter, in the next two Sections we report results separately for the experiments using ML-kNN and BPMLL as the classifier. In addition, in each Section we report in a separate table the results for the small datasets (with less than 300 features), where all original features were given to each feature selection method; and report in separate tables the results for other large datasets (with more than 1,000 features), where a univariate filter method was applied in a pre-processing phase to reduce the feature space size.

4.5.2 Experimental Results for gmiML-CFS and Other Multi-Label Feature Selection Methods Using the ML-kNN Classifier

Clearly, in Tables 4.28 - 4.32, gmiML-CFS obtained substantially better predictive accuracy (substantially lower overall average rank) across all datasets and all accuracy measures than NoFS, BR and RFML in most cases. In Table 4.28, reporting results for small datasets, gmiML-CFS obtained the same average rank as CFS-U (1.9); while NoFS, BR and RFML obtained substantially larger average ranks (2.5, 3.3 and 3.9, respectively). Moreover, gmiML-CFS outperforms CFS-U according to three different predictive accuracy measures: Coverage, OneErr and R-Loss.

Tables 4.29 - 4.32 report results for the large datasets, with the feature space size varying from 100 to 400. In Table 4.29, when the feature space size equals to 100, CFS-U obtained the best overall average rank (1.9); while gmiML-CFS and NoFS jointly obtained the second best overall average rank 2.3 and outperform BR and RFML, which obtained overall average rank 4.6 and 3.9, respectively.

In Table 4.30, the best method was CFS-U, with an overall average rank of

Table 4.28: Values of five multi-label predictive accuracy measures for gmiML-CFS and other feature selection methods using MLkNN as the classifier - small datasets

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
CAL500	NoFS	0.490	4.0	130.950	4.0	0.139	2.0	0.120	2.5	0.184	3.0	3.1
	BR(kNN)	0.502	1.0	129.738	2.0	0.251	5.0	0.142	5.0	0.181	2.0	3.0
	CFS-U	0.500	2.0	129.260	1.0	0.130	1.0	0.140	4.0	0.178	1.0	1.8
	RFML	0.486	5.0	130.234	3.0	0.139	4.0	0.116	1.0	0.185	5.0	3.6
	gmi-ML-CFS	0.492	3.0	131.068	3.0	0.139	1.0	0.120	1.0	0.185	1.0	1.8
Scene	NoFS	0.540	1.0	1.960	2.0	0.235	4.0	0.667	1.0	0.370	2.0	2.0
	BR(kNN)	0.531	2.0	1.793	1.0	0.257	5.0	0.722	4.0	0.341	1.0	2.6
	CFS-U	0.528	3.0	2.066	4.0	0.198	2.0	0.668	2.0	0.392	4.0	3.0
	RFML	0.506	4.0	2.028	3.0	0.192	1.0	0.727	5.0	0.389	3.0	3.2
	gmi-ML-CFS	0.499	3.0	2.161	3.0	0.210	2.0	0.715	3.0	0.411	3.0	2.8
Emotions	NoFS	0.797	4.0	1.876	2.0	0.209	2.0	0.282	2.5	0.159	1.0	2.3
	BR(kNN)	0.800	2.0	1.853	1.0	0.211	3.0	0.291	4.0	0.162	3.0	2.6
	CFS-U	0.808	1.0	1.891	3.0	0.196	1.0	0.248	1.0	0.161	2.0	1.6
	RFML	0.758	5.0	2.104	5.0	0.262	5.0	0.347	5.0	0.203	5.0	5.0
	gmi-ML-CFS	0.800	1.0	1.921	1.0	0.215	2.0	0.282	3.0	0.174	2.0	1.8
Yeast	NoFS	0.757	2.0	6.364	2.0	0.198	2.0	0.242	4.0	0.171	2.0	2.4
	BR(kNN)	0.741	5.0	6.610	5.0	0.226	5.0	0.257	5.0	0.189	5.0	5.0
	CFS-U	0.761	1.0	6.341	1.0	0.196	1.0	0.237	2.0	0.169	1.0	1.2
	RFML	0.749	4.0	6.543	4.0	0.205	4.0	0.240	3.0	0.183	4.0	3.8
	gmi-ML-CFS	0.756	1.0	6.495	1.0	0.204	1.0	0.230	1.0	0.176	1.0	1.0
MEAN	NoFS		2.8		2.5		2.5		2.5		2.0	2.5
	BR(kNN)		2.5		2.3		4.5		4.5		2.8	3.3
	CFS-U		1.8		2.3		1.3		2.3		2.0	1.9
	RFML		4.5		3.8		3.5		3.5		4.3	3.9
	gmi-ML-CFS		2.0		2.0		1.5		2.0		1.8	1.9

1.7. The second best method was gmiML-CFS, which outperforms NoFS, BR and RFML with overall average rank = 2.3.

In Table 4.31 the best methods were CFS-U and gmiML-CFS, both with an overall average rank of 1.8. These two methods outperform NoFS, BR and RFML, which obtained overall average rank = 2.8, 4.8 and 3.7, respectively.

In table 4.32 the best method was CFS-U, with an overall average rank of 1.8; while the second best method was gmiML-CFS, which outperformed NoFS, BR and RFML with overall average Rank = 1.9.

Table 4.33 reports the summary of results in terms of the overall average ranking and the number of selected features of gmiML-CFSabs and multi-label feature selection approaches when using MLkNN as the classifier. Like in previous Sec-

Table 4.29: Values of five multi-label predictive accuracy measures for gmiML-CFS and other feature selection methods using MLkNN as the classifier - feature space size = 100

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Business	NoFS	0.874	3.0	2.369	1.0	0.028	2.0	0.124	3.0	0.043	1.0	2.0
	BR(kNN)	0.854	5.0	2.725	5.0	0.042	5.0	0.139	5.0	0.048	5.0	5.0
	CFS-U	0.875	1.0	2.379	3.0	0.028	1.0	0.122	1.0	0.043	2.5	1.7
	RFML	0.867	4.0	2.467	4.0	0.029	4.0	0.132	4.0	0.045	4.0	4.0
	gmi-ML-CFS	0.874	2.0	2.371	2.0	0.028	3.0	0.123	2.0	0.043	2.5	2.3
Art	NoFS	0.529	2.0	5.306	2.0	0.059	3.0	0.592	3.0	0.146	2.0	2.4
	BR(kNN)	0.432	5.0	5.971	5.0	0.229	5.0	0.752	5.0	0.176	5.0	5.0
	CFS-U	0.533	1.0	5.272	1.0	0.059	2.0	0.586	1.0	0.145	1.0	1.2
	RFML	0.466	4.0	5.911	4.0	0.062	4.0	0.689	4.0	0.168	4.0	4.0
	gmi-ML-CFS	0.528	3.0	5.398	3.0	0.059	1.0	0.588	2.0	0.150	3.0	2.4
Education	NoFS	0.543	2.5	3.938	2.0	0.041	1.5	0.602	2.0	0.093	3.0	2.2
	BR(kNN)	0.476	5.0	4.645	5.0	0.145	5.0	0.681	5.0	0.110	5.0	5.0
	CFS-U	0.545	1.0	3.921	1.0	0.041	1.5	0.597	1.0	0.092	1.0	1.1
	RFML	0.486	4.0	4.421	4.0	0.045	4.0	0.678	4.0	0.107	4.0	4.0
	ML-CFS	0.543	2.5	3.982	3.0	0.042	3.0	0.603	3.0	0.093	2.0	2.7
Recreation	NoFS	0.536	1.0	4.333	2.0	0.058	1.0	0.595	1.0	0.157	2.0	1.4
	BR(kNN)	0.376	5.0	5.603	4.0	0.346	5.0	0.805	5.0	0.222	5.0	4.8
	CFS-U	0.535	3.0	4.302	1.0	0.059	3.0	0.598	2.0	0.157	1.0	2.0
	RFML	0.385	4.0	5.665	5.0	0.065	4.0	0.795	4.0	0.218	4.0	4.2
	gmi-ML-CFS	0.535	2.0	4.349	3.0	0.059	2.0	0.601	3.0	0.159	3.0	2.6
Health	NoFS	0.631	3.0	3.784	3.0	0.049	1.5	0.476	1.0	0.075	2.0	2.1
	BR(kNN)	0.616	5.0	4.062	5.0	0.129	5.0	0.489	5.0	0.078	4.0	4.8
	CFS-U	0.632	2.0	3.767	2.0	0.049	1.5	0.477	3.0	0.075	1.0	1.9
	RFML	0.624	4.0	3.900	4.0	0.050	4.0	0.482	4.0	0.078	5.0	4.2
	gmi-ML-CFS	0.634	1.0	3.747	1.0	0.049	3.0	0.476	2.0	0.075	3.0	2.0
Enter.ment	NoFS	0.597	1.0	3.135	1.0	0.056	3.0	0.537	1.0	0.116	1.0	1.4
	BR(kNN)	0.465	5.0	3.984	5.0	0.281	5.0	0.715	5.0	0.159	5.0	5.0
	CFS-U	0.583	3.0	3.194	3.0	0.055	1.0	0.548	3.0	0.118	2.0	2.4
	RFML	0.491	4.0	3.920	4.0	0.064	4.0	0.678	4.0	0.151	4.0	4.0
	gmi-ML-CFS	0.593	2.0	3.158	2.0	0.056	2.0	0.548	2.0	0.119	3.0	2.2
Computer	NoFS	0.630	2.0	4.289	1.0	0.040	2.5	0.443	2.0	0.091	2.0	1.9
	BR(kNN)	0.599	5.0	4.840	5.0	0.112	5.0	0.475	5.0	0.101	5.0	5.0
	CFS-U	0.631	1.0	4.291	2.0	0.040	2.5	0.442	1.0	0.091	1.0	1.5
	RFML	0.610	4.0	4.533	4.0	0.042	4.0	0.471	4.0	0.097	4.0	4.0
	gmi-ML-CFS	0.623	3.0	4.416	3.0	0.040	1.0	0.450	3.0	0.094	3.0	2.6
Science	NoFS	0.456	3.0	6.852	2.0	0.035	3.0	0.676	3.0	0.134	2.0	2.6
	BR(kNN)	0.391	5.0	8.112	5.0	0.236	5.0	0.758	5.0	0.165	5.0	5.0
	CFS-U	0.462	2.0	6.812	1.0	0.035	2.0	0.668	2.0	0.133	1.0	1.6
	RFML	0.418	4.0	7.248	4.0	0.036	4.0	0.724	4.0	0.143	4.0	4.0
	gmi-ML-CFS	0.463	1.0	6.965	3.0	0.034	1.0	0.662	1.0	0.137	3.0	1.8
Enron	NoFS	0.584	2.0	13.380	1.0	0.058	3.0	0.396	3.5	0.097	1.0	2.1
	BR(kNN)	0.547	5.0	14.109	5.0	0.098	5.0	0.413	5.0	0.106	5.0	5.0
	CFS-U	0.587	1.0	13.501	2.0	0.057	2.0	0.390	2.0	0.098	2.0	1.8
	RFML	0.580	4.0	13.883	4.0	0.059	4.0	0.396	3.5	0.103	4.0	3.9
	gmi-ML-CFS	0.583	3.0	13.679	3.0	0.057	1.0	0.389	1.0	0.100	3.0	2.2
Medical	NoFS	0.717	5.0	3.614	5.0	0.019	5.0	0.374	5.0	0.062	5.0	5.0
	BR(kNN)	0.796	1.0	2.299	1.0	0.017	3.0	0.281	1.0	0.037	1.0	1.4
	CFS-U	0.758	4.0	3.505	4.0	0.018	4.0	0.301	3.5	0.059	4.0	3.9
	RFML	0.765	2.0	3.461	3.0	0.017	2.0	0.299	2.0	0.057	3.0	2.4
	gmi-ML-CFS	0.760	3.0	3.372	2.0	0.017	1.0	0.301	3.5	0.055	2.0	2.3
MEAN	NoFS		2.5		2.0		2.6		2.5		2.1	2.3
	BR(kNN)		4.6		4.5		4.8		4.6		4.5	4.6
	CFS-U		1.9		2.0		2.1		2.0		1.7	1.9
	RFML		3.8		4.0		3.8		3.8		4.0	3.9
	gmi-ML-CFS		2.3		2.5		1.8		2.3		2.8	2.3

Table 4.30: Values of five multi-label predictive accuracy measures for gmiML-CFS and other feature selection methods using MLkNN as the classifier - feature space size = 200

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Business	NoFS	0.874	3.0	2.369	1.0	0.028	2.0	0.124	3.0	0.043	1.0	2.0
	BR(kNN)	0.853	5.0	2.726	5.0	0.042	5.0	0.139	5.0	0.049	5.0	5.0
	CFS-U	0.877	1.0	2.262	1.0	0.028	2.0	0.124	1.0	0.040	1.0	1.2
	RFML	0.863	4.0	2.488	4.0	0.029	4.0	0.137	4.0	0.046	4.0	4.0
	gmi-ML-CFS	0.873	3.0	2.357	3.0	0.028	2.0	0.124	3.0	0.042	3.0	2.8
Art	NoFS	0.519	3.0	5.319	2.0	0.059	3.0	0.605	3.0	0.147	2.0	2.6
	BR(kNN)	0.414	5.0	7.523	5.0	0.557	5.0	0.752	5.0	0.226	5.0	5.0
	CFS-U	0.541	1.0	5.190	1.0	0.058	1.0	0.572	1.0	0.141	1.0	1.0
	RFML	0.452	4.0	5.959	4.0	0.064	4.0	0.712	4.0	0.171	4.0	4.0
	gmi-ML-CFS	0.537	2.0	5.323	3.0	0.059	2.0	0.579	2.0	0.148	3.0	2.4
Education	NoFS	0.544	3.0	3.895	2.0	0.041	3.0	0.602	3.0	0.092	2.0	2.6
	BR(kNN)	0.467	5.0	5.435	5.0	0.271	5.0	0.681	5.0	0.125	5.0	5.0
	CFS-U	0.549	2.0	3.876	1.0	0.041	1.0	0.592	2.0	0.091	1.0	1.4
	RFML	0.486	4.0	4.426	4.0	0.044	4.0	0.678	4.0	0.107	4.0	4.0
	ML-CFS	0.551	1.0	3.918	3.0	0.041	2.0	0.583	1.0	0.092	3.0	2.0
Recreation	NoFS	0.553	3.0	4.321	3.0	0.056	3.0	0.570	3.0	0.158	3.0	3.0
	BR(kNN)	0.314	5.0	7.562	5.0	0.559	5.0	0.803	5.0	0.311	5.0	5.0
	CFS-U	0.571	2.0	4.166	1.0	0.055	2.0	0.540	1.0	0.152	1.0	1.4
	RFML	0.407	4.0	5.172	4.0	0.065	4.0	0.765	4.0	0.199	4.0	4.0
	gmi-ML-CFS	0.572	1.0	4.223	2.0	0.054	1.0	0.540	2.0	0.152	2.0	1.6
Health	NoFS	0.673	3.0	3.453	3.0	0.044	3.0	0.412	3.0	0.065	3.0	3.0
	BR(kNN)	0.607	5.0	4.037	5.0	0.158	5.0	0.489	5.0	0.081	5.0	5.0
	CFS-U	0.684	2.0	3.380	1.0	0.043	2.0	0.402	2.0	0.063	1.0	1.6
	RFML	0.667	4.0	3.578	4.0	0.045	4.0	0.422	4.0	0.068	4.0	4.0
	gmi-ML-CFS	0.685	1.0	3.400	2.0	0.042	1.0	0.392	1.0	0.063	2.0	1.4
Enter.ment	NoFS	0.624	1.0	2.982	1.0	0.056	3.0	0.500	1.0	0.108	1.0	1.4
	BR(kNN)	0.451	5.0	4.843	5.0	0.460	5.0	0.715	5.0	0.192	5.0	5.0
	CFS-U	0.613	2.0	3.049	2.0	0.054	2.0	0.513	3.0	0.111	2.0	2.2
	RFML	0.501	4.0	3.762	4.0	0.064	4.0	0.669	4.0	0.145	4.0	4.0
	gmi-ML-CFS	0.604	3.0	3.117	3.0	0.054	1.0	0.513	2.0	0.113	3.0	2.4
Computer	NoFS	0.647	2.0	4.125	2.0	0.038	2.0	0.424	2.0	0.087	1.5	1.9
	BR(kNN)	0.589	5.0	5.099	5.0	0.160	5.0	0.475	5.0	0.110	5.0	5.0
	CFS-U	0.648	1.0	4.115	1.0	0.038	1.0	0.423	1.0	0.087	1.5	1.1
	RFML	0.619	4.0	4.408	4.0	0.041	4.0	0.456	4.0	0.094	4.0	4.0
	gmi-ML-CFS	0.638	3.0	4.181	3.0	0.039	3.0	0.436	3.0	0.089	3.0	3.0
Science	NoFS	0.476	3.0	6.617	2.0	0.034	3.0	0.654	3.0	0.129	2.0	2.6
	BR(kNN)	0.386	5.0	8.877	5.0	0.490	5.0	0.758	5.0	0.182	5.0	5.0
	CFS-U	0.487	1.0	6.563	1.0	0.034	1.0	0.640	2.0	0.127	1.0	1.2
	RFML	0.437	4.0	7.131	4.0	0.036	4.0	0.702	4.0	0.141	4.0	4.0
	gmi-ML-CFS	0.484	2.0	6.808	3.0	0.034	2.0	0.638	1.0	0.133	3.0	2.2
Enron	NoFS	0.596	1.0	13.404	2.0	0.057	1.0	0.373	1.0	0.097	2.0	1.4
	BR(kNN)	0.566	5.0	14.288	5.0	0.094	5.0	0.413	5.0	0.103	5.0	5.0
	CFS-U	0.589	2.0	13.325	1.0	0.058	3.0	0.383	2.0	0.096	1.0	1.8
	RFML	0.578	4.0	13.636	3.0	0.058	2.0	0.406	4.0	0.101	4.0	3.4
	gmi-ML-CFS	0.580	3.0	13.719	4.0	0.059	4.0	0.396	3.0	0.101	3.0	3.4
Medical	NoFS	0.745	5.0	3.557	5.0	0.019	5.0	0.321	5.0	0.060	5.0	5.0
	BR(kNN)	0.825	1.0	2.228	1.0	0.016	2.0	0.231	1.5	0.033	1.0	1.3
	CFS-U	0.769	4.0	3.242	4.0	0.018	4.0	0.292	4.0	0.053	4.0	4.0
	RFML	0.805	3.0	2.892	2.0	0.017	3.0	0.257	3.0	0.044	2.0	2.6
	gmi-ML-CFS	0.807	2.0	2.929	3.0	0.016	1.0	0.231	1.5	0.046	3.0	2.1
MEAN	NoFS		2.6		2.4		2.8		2.6		2.4	2.6
	BR(kNN)		4.6		4.6		4.7		4.7		4.6	4.6
	CFS-U		1.8		1.4		1.9		1.9		1.5	1.7
	RFML		3.9		3.7		3.7		3.9		3.8	3.8
	gmi-ML-CFS		2.1		2.9		1.9		2.0		2.8	2.3

Table 4.31: Values of five multi-label predictive accuracy measures for gmiML-CFS and other feature selection methods using MLkNN as the classifier - feature space size = 300

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Business	NoFS	0.876	2.0	2.288	2.0	0.028	1.0	0.124	2.0	0.041	3.0	2.0
	BR(kNN)	0.854	5.0	2.736	5.0	0.044	5.0	0.139	4.0	0.049	5.0	4.8
	CFS-U	0.877	1.0	2.280	1.0	0.028	2.0	0.123	1.0	0.040	1.5	1.3
	RFML	0.862	4.0	2.456	4.0	0.030	4.0	0.140	5.0	0.045	4.0	4.2
	gmi-ML-CFS	0.876	3.0	2.292	3.0	0.028	3.0	0.127	3.0	0.040	1.5	2.7
Art	NoFS	0.521	3.0	5.256	2.0	0.060	3.0	0.607	3.0	0.144	2.0	2.6
	BR(kNN)	0.234	5.0	8.567	5.0	0.627	5.0	0.978	5.0	0.269	5.0	5.0
	CFS-U	0.543	1.0	5.108	1.0	0.059	2.0	0.572	1.0	0.139	1.0	1.2
	RFML	0.453	4.0	5.926	4.0	0.064	4.0	0.716	4.0	0.169	4.0	4.0
	gmi-ML-CFS	0.540	2.0	5.278	3.0	0.058	1.0	0.575	2.0	0.145	3.0	2.2
Education	NoFS	0.541	3.0	3.914	3.0	0.041	3.0	0.604	3.0	0.092	3.0	3.0
	BR(kNN)	0.151	5.0	10.153	5.0	0.470	5.0	0.987	5.0	0.284	5.0	5.0
	CFS-U	0.548	2.0	3.877	1.0	0.041	2.0	0.596	2.0	0.091	1.0	1.6
	RFML	0.491	4.0	4.347	4.0	0.044	4.0	0.669	4.0	0.105	4.0	4.0
	ML-CFS	0.552	1.0	3.895	2.0	0.041	1.0	0.588	1.0	0.091	2.0	1.4
Recreation	NoFS	0.552	3.0	4.296	3.0	0.056	3.0	0.573	3.0	0.157	3.0	3.0
	BR(kNN)	0.154	5.0	9.750	5.0	0.674	5.0	0.995	5.0	0.414	5.0	5.0
	CFS-U	0.576	2.0	4.074	1.0	0.055	2.0	0.542	2.0	0.147	1.0	1.6
	RFML	0.421	4.0	4.977	4.0	0.065	4.0	0.754	4.0	0.191	4.0	4.0
	gmi-ML-CFS	0.581	1.0	4.147	2.0	0.054	1.0	0.530	1.0	0.150	2.0	1.4
Health	NoFS	0.674	3.0	3.441	3.0	0.045	3.0	0.418	3.0	0.065	3.0	3.0
	BR(kNN)	0.602	5.0	4.386	5.0	0.220	5.0	0.489	5.0	0.089	5.0	5.0
	CFS-U	0.682	2.0	3.373	2.0	0.044	2.0	0.407	2.0	0.063	2.0	2.0
	RFML	0.660	4.0	3.603	4.0	0.046	4.0	0.429	4.0	0.068	4.0	4.0
	gmi-ML-CFS	0.699	1.0	3.303	1.0	0.042	1.0	0.380	1.0	0.061	1.0	1.0
Enter.ment	NoFS	0.608	3.0	3.034	3.0	0.057	3.0	0.523	3.0	0.111	3.0	3.0
	BR(kNN)	0.211	5.0	7.262	5.0	0.513	5.0	0.923	5.0	0.324	5.0	5.0
	CFS-U	0.612	2.0	2.975	1.0	0.055	2.0	0.517	2.0	0.108	1.0	1.6
	RFML	0.510	4.0	3.666	4.0	0.063	4.0	0.663	4.0	0.142	4.0	4.0
	gmi-ML-CFS	0.627	1.0	3.004	2.0	0.054	1.0	0.494	1.0	0.110	2.0	1.4
Computer	NoFS	0.651	1.0	4.086	2.0	0.037	2.0	0.423	1.0	0.086	2.0	1.6
	BR(kNN)	0.251	5.0	8.628	5.0	0.507	5.0	0.939	5.0	0.205	5.0	5.0
	CFS-U	0.651	2.0	4.067	1.0	0.037	1.0	0.424	2.0	0.086	1.0	1.4
	RFML	0.625	4.0	4.359	4.0	0.040	4.0	0.450	4.0	0.092	4.0	4.0
	gmi-ML-CFS	0.646	3.0	4.161	3.0	0.038	3.0	0.427	3.0	0.088	3.0	3.0
Science	NoFS	0.475	3.0	6.611	2.0	0.034	2.0	0.660	3.0	0.130	3.0	2.6
	BR(kNN)	0.119	5.0	14.552	5.0	0.559	5.0	0.967	5.0	0.332	5.0	5.0
	CFS-U	0.477	2.0	6.535	1.0	0.035	3.0	0.657	2.0	0.128	1.0	1.8
	RFML	0.423	4.0	7.242	4.0	0.036	4.0	0.712	4.0	0.145	4.0	4.0
	gmi-ML-CFS	0.489	1.0	6.622	3.0	0.034	1.0	0.629	1.0	0.129	2.0	1.6
Enron	NoFS	0.567	3.0	13.629	3.0	0.059	4.0	0.404	3.0	0.100	3.0	3.2
	BR(kNN)	0.554	5.0	14.808	5.0	0.147	5.0	0.508	5.0	0.113	5.0	5.0
	CFS-U	0.567	4.0	13.584	2.0	0.058	2.5	0.396	2.0	0.100	2.0	2.5
	RFML	0.581	2.0	13.884	4.0	0.058	2.5	0.389	1.0	0.102	4.0	2.7
	gmi-ML-CFS	0.581	1.0	13.432	1.0	0.058	1.0	0.406	4.0	0.098	1.0	1.6
Medical	NoFS	0.738	4.0	3.578	5.0	0.019	4.0	0.336	4.0	0.060	5.0	4.4
	BR(kNN)	0.694	5.0	2.816	1.0	0.028	5.0	0.411	5.0	0.047	2.0	3.6
	CFS-U	0.776	3.0	3.222	4.0	0.018	3.0	0.292	3.0	0.052	4.0	3.4
	RFML	0.805	2.0	2.983	3.0	0.015	1.0	0.248	2.0	0.047	3.0	2.2
	gmi-ML-CFS	0.819	1.0	2.831	2.0	0.016	2.0	0.225	1.0	0.044	1.0	1.4
MEAN	NoFS		2.8		2.8		2.8		2.8		3.0	2.8
	BR(kNN)		5.0		4.6		5.0		4.9		4.7	4.8
	CFS-U		2.1		1.5		2.2		1.9		1.6	1.8
	RFML		3.6		3.9		3.6		3.6		3.9	3.7
	gmi-ML-CFS		1.5		2.2		1.5		1.8		1.9	1.8

Table 4.32: Values of five multi-label predictive accuracy measures for gmiML-CFS and other feature selection methods using MLkNN as the classifier - feature space size = 400

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Business	NoFS	0.881	1.0	2.26	2.0	0.028	1.0	0.119	1.0	0.039	1.0	1.2
	BR(kNN)	0.767	5.0	4.01	5.0	0.294	5.0	0.139	5.0	0.075	5.0	5.0
	CFS-U	0.879	2.0	2.24	1.0	0.028	2.0	0.123	2.5	0.039	2.0	1.9
	RFML	0.865	4.0	2.454	4.0	0.029	4.0	0.137	4.0	0.045	4.0	4.0
	gmi-ML-CFS	0.877	3.0	2.299	3.0	0.028	3.0	0.123	2.5	0.041	3.0	2.9
Art	NoFS	0.509	3.0	5.34	3.0	0.060	3.0	0.631	3.0	0.147	3.0	3.0
	BR(kNN)	0.150	5.0	12.52	5.0	0.468	5.0	0.980	5.0	0.424	5.0	5.0
	CFS-U	0.546	1.0	5.08	1.0	0.058	2.0	0.569	1.0	0.137	1.0	1.2
	RFML	0.461	4.0	5.885	4.0	0.064	4.0	0.700	4.0	0.166	4.0	4.0
	gmi-ML-CFS	0.536	2.0	5.298	2.0	0.057	1.0	0.582	2.0	0.146	2.0	1.8
Education	NoFS	0.535	3.0	3.95	3.0	0.042	3.0	0.611	3.0	0.093	3.0	3.0
	BR(kNN)	0.143	5.0	9.95	5.0	0.508	5.0	0.999	5.0	0.272	5.0	5.0
	CFS-U	0.555	2.0	3.78	1.0	0.041	2.0	0.589	2.0	0.089	1.0	1.6
	RFML	0.487	4.0	4.382	4.0	0.044	4.0	0.674	4.0	0.106	4.0	4.0
	ML-CFS	0.559	1.0	3.848	2.0	0.040	1.0	0.574	1.0	0.090	2.0	1.4
Recreation	NoFS	0.552	3.0	4.24	3.0	0.057	3.0	0.576	3.0	0.155	3.0	3.0
	BR(kNN)	0.176	5.0	10.93	5.0	0.684	5.0	0.949	5.0	0.452	5.0	5.0
	CFS-U	0.578	2.0	4.06	2.0	0.055	2.0	0.539	2.0	0.147	2.0	2.0
	RFML	0.430	4.0	4.820	4.0	0.064	4.0	0.740	4.0	0.185	4.0	4.0
	gmi-ML-CFS	0.590	1.0	4.052	1.0	0.054	1.0	0.517	1.0	0.146	1.0	1.0
Health	NoFS	0.692	3.0	3.30	3.0	0.043	3.0	0.395	3.0	0.061	3.0	3.0
	BR(kNN)	0.378	5.0	5.17	5.0	0.303	5.0	0.957	5.0	0.114	5.0	5.0
	CFS-U	0.701	2.0	3.25	2.0	0.043	2.0	0.378	2.0	0.060	2.0	2.0
	RFML	0.674	4.0	3.514	4.0	0.044	4.0	0.412	4.0	0.067	4.0	4.0
	gmi-ML-CFS	0.721	1.0	3.193	1.0	0.040	1.0	0.348	1.0	0.058	1.0	1.0
Enter.ment	NoFS	0.617	3.0	3.00	2.0	0.057	3.0	0.510	3.0	0.110	3.0	2.8
	BR(kNN)	0.221	5.0	6.82	5.0	0.567	5.0	0.961	5.0	0.297	5.0	5.0
	CFS-U	0.630	1.0	2.89	1.0	0.054	2.0	0.495	2.0	0.105	1.0	1.4
	RFML	0.520	4.0	3.581	4.0	0.064	4.0	0.649	4.0	0.137	4.0	4.0
	gmi-ML-CFS	0.624	2.0	3.002	3.0	0.053	1.0	0.491	1.0	0.109	2.0	1.8
Computer	NoFS	0.655	2.0	4.03	2.0	0.037	2.0	0.418	2.0	0.084	2.0	2.0
	BR(kNN)	0.213	5.0	8.45	5.0	0.584	5.0	0.967	5.0	0.213	5.0	5.0
	CFS-U	0.655	1.0	4.01	1.0	0.037	1.0	0.417	1.0	0.084	1.0	1.0
	RFML	0.628	4.0	4.315	4.0	0.040	4.0	0.448	4.0	0.092	4.0	4.0
	gmi-ML-CFS	0.646	3.0	4.184	3.0	0.038	3.0	0.429	3.0	0.088	3.0	3.0
Science	NoFS	0.462	3.0	6.68	2.0	0.035	3.0	0.671	3.0	0.132	2.0	2.6
	BR(kNN)	0.145	5.0	13.28	5.0	0.593	5.0	0.980	5.0	0.293	5.0	5.0
	CFS-U	0.482	2.0	6.53	1.0	0.034	2.0	0.648	2.0	0.128	1.0	1.6
	RFML	0.434	4.0	7.101	4.0	0.036	4.0	0.703	4.0	0.141	4.0	4.0
	gmi-ML-CFS	0.485	1.0	6.741	3.0	0.034	1.0	0.629	1.0	0.132	3.0	1.8
Enron	NoFS	0.583	1.0	13.40	1.0	0.056	1.0	0.382	1.0	0.098	1.0	1.0
	BR(kNN)	0.471	5.0	14.22	5.0	0.165	5.0	0.760	5.0	0.113	5.0	5.0
	CFS-U	0.580	2.0	13.47	2.0	0.057	2.0	0.385	2.0	0.099	2.0	2.0
	RFML	0.579	3.0	13.814	4.0	0.058	3.0	0.392	3.0	0.102	4.0	3.4
	gmi-ML-CFS	0.575	4.0	13.624	3.0	0.059	4.0	0.406	4.0	0.100	3.0	3.6
Medical	NoFS	0.728	4.0	3.72	4.0	0.020	4.0	0.349	4.0	0.063	4.0	4.0
	BR(kNN)	0.110	5.0	13.81	5.0	0.420	5.0	0.980	5.0	0.291	5.0	5.0
	CFS-U	0.768	3.0	3.34	3.0	0.019	3.0	0.295	3.0	0.055	3.0	3.0
	RFML	0.810	2.0	3.005	2.0	0.017	2.0	0.226	2.0	0.047	2.0	2.0
	gmi-ML-CFS	0.819	1.0	2.812	1.0	0.016	1.0	0.225	1.0	0.044	1.0	1.0
MEAN	NoFS		2.6		2.5		2.6		2.6		2.5	2.6
	BR(kNN)		5.0		5.0		5.0		5.0		5.0	5.0
	CFS-U		1.8		1.5		2.0		2.0		1.6	1.8
	RFML		3.7		3.8		3.7		3.7		3.8	3.7
	gmi-ML-CFS		1.9		2.2		1.7		1.8		2.1	1.9

Table 4.33: Summary of results in terms of average ranking (AR) and the number of selected features (Sel.F) of gmiML-CFS and other multi-label feature selection methods using ML-kNN as the classifier

Datasets and feature space size	NoFS		BR(kNN)		CFS-U		RFML		gmi-ML-CFS	
	AR	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**
Emotion	2.30	72.00	2.60	72.00	1.60	52.00	5.00	10.00	1.80	10.00
CAL500	3.10	68.00	3.00	68.00	1.80	51.00	3.60	12.90	1.80	12.90
Scene	2.00	294.00	2.60	294.00	3.00	234.00	3.20	36.00	2.80	36.00
Yeast	2.40	103.00	5.00	103.00	1.20	74.00	3.80	22.00	1.00	22.00
100	2.31	100.00	4.60	100.00	1.91	73.90	3.87	22.40	2.31	22.40
200	2.55	200.00	4.63	200.00	1.69	128.40	3.80	34.30	2.33	34.30
300	2.84	300.00	4.84	300.00	1.84	174.80	3.71	44.10	1.77	44.10
400	2.56	400.00	5.00	400.00	1.77	214.40	3.74	57.00	1.93	57.00

tions, the results in the last four rows are an average over results for all datasets, for each feature space size.

In Table 4.33, CFS-U obtains the best average rank with the larger selected feature subset when compared with RFML and gmiML-CFS in general. For example; in CAL500, CFS-U selects a feature subset about four times larger than the one selected by gmiML-CFS (51 and 12.90 features, respectively). The difference between the average ranks of CFS-U and gmiML-CFS was small in most cases (between 0.04 and 0.4) except when the feature space sizes was equal to 200: the difference is 0.64 (2.33 - 1.69). Moreover, RFML, which has the same size of selected feature subset as gmiML-CFS, obtains much worse average rank when compared with gmiML-CFS; while NoFS and BR, which use either the full set of features for small datasets or the feature subset selected by the univariate approach (original feature space size), still obtain a larger average rank than gmiML-CFS.

Figure 4.1 shows the overall average ranking (AR) for gmiML-CFS and the other multi-label feature selection methods plotted against the average size of selected features across all dataset and feature space sizes, when using ML-kNN as the classifier. Clearly, gmi-ML-CFS occupies a very good position in this plot. Its average ranking is just slightly worse than the one of CFS-U, but gmi-ML-CFS is much more to the left (selects a smaller number of features) than CFS-U. In

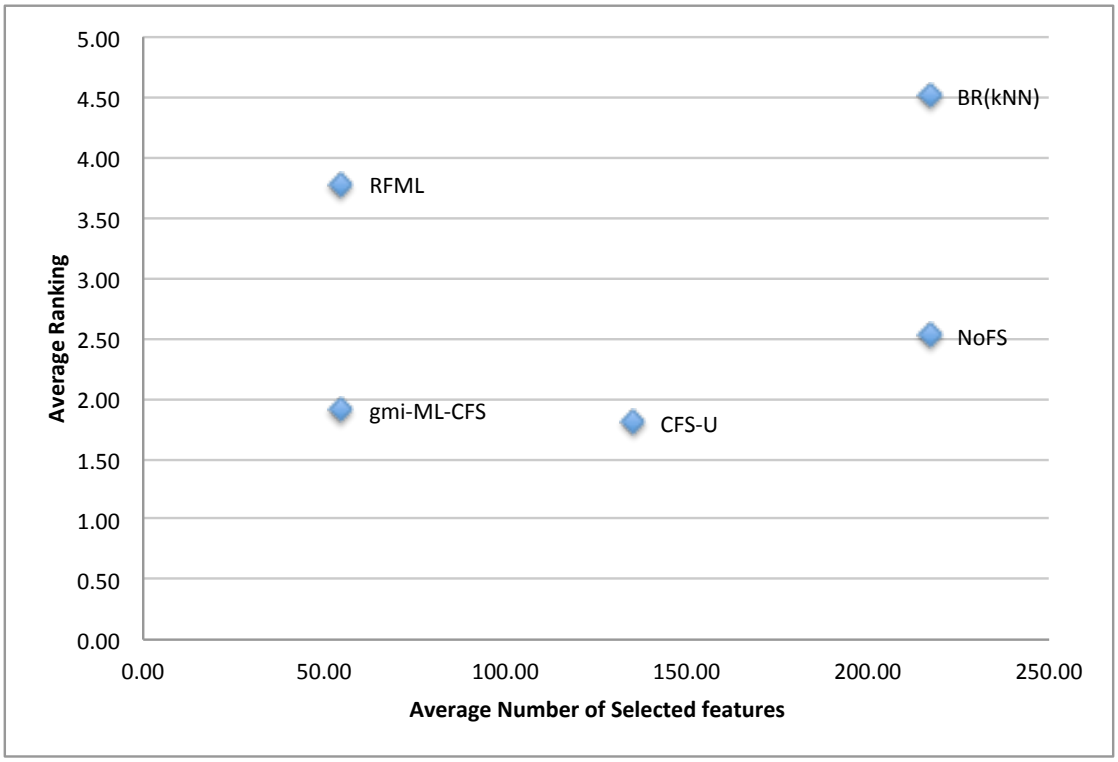


Figure 4.1: Overall average ranking (AR) for gmiML-CFS and the other multi-label feature selection methods plotted against the average size of selected features across all datasets and feature space sizes, when using ML-kNN as the classifier

addition gmi-ML-CFS and RFML are in the same position along axis X (number of selected features), but gmi-ML-CFS is in a much lower position along the Y axis (better average ranking) than RFML. That is, the graph shows that gmi-ML-CFS achieves a very good trade-off between predictive accuracy and number of selected features.

In general, gmiML-CFS selected the smallest feature subset while obtaining the second best predictive accuracy out of five different multi-label feature selection approaches.

Table 4.34 presents a summary of the results from another perspective, reporting the average ranks (in terms of predictive accuracy) for each dataset - i.e, averaging over different feature space sizes. In the first four rows (for small

Table 4.34: Summary of overall average ranking (AR) for gmiML-CFS and other multi-label feature selection methods across four feature space sizes using ML-kNN as the classifier

Dataset	**Overall Average Rank (AR) across 4 feature space sizes**				
	NoFS	**BR(kNN)**	**CFS-U**	**RFML**	**gmi-ML-CFS**
CAL500	3.1(4)	3(3)	1.8(1)	3.6(5)	1.8(1)
Scene	2(1)	2.6(2)	3(4)	3.2(5)	2.8(3)
Emotions	2.3(3)	2.6(4)	1.6(1)	5(5)	1.8(2)
Yeast	2.4(3)	5(5)	1.2(2)	3.8(4)	1(1)
Enron	1(1)	5(5)	2(2)	3.4(3)	3.6(4)
Medical	4(4)	5(5)	3(3)	2(2)	1(1)
Business	1.2(1)	5(5)	1.9(2)	4(4)	2.9(3)
Art	3(3)	5(5)	1.2(1)	4(4)	1.8(2)
Education	3(3)	5(5)	1.6(2)	4(4)	1.4(1)
Recreation	3(3)	5(5)	2(2)	4(4)	1(1)
Health	3(3)	5(5)	2(2)	4(4)	1(1)
Ent.ment	2.8(3)	5(5)	1.4(1)	4(4)	1.8(2)
Computer	2(2)	5(5)	1(1)	4(4)	3(3)
Science	2.6(3)	5(5)	1.6(1)	4(4)	1.8(2)
Average	2.53(2.64)	4.51(4.57)	1.81(1.79)	3.79(4)	1.91(1.93)

datasets) in each cell the first value is taken directly from Table 4.28. For the other rows (large datasets), in each cell of the table, the first value is the average rank computed by averaging the corresponding ranks in Tables 4.29 - 4.32. In all cells of the table, the value between brackets is the "rank of the average ranks". This latter value was used for the statistical test of significance mentioned next.

Using the results shows in Table 4.34, we ran the Friedman test and confidently conclude that there is a significant difference among the 5 methods on the 14 evaluation datasets at the 0.05 level of significance for a two tailed test (p value $<$ 0.00001). Running the Holm's posthoc test on these data using gmiGA-wrap as the control method, there are no significant differences when comparing gmiML-CFS versus CFS-U and NoFS at the 0.05 significance level, but there is a significant difference between gmiML-CFS versus BR, as well as between gmiML-CFS and and RFML at the same level of significance (p value = 0.00012 and 0.00461, respectively).

4.5.3 Experimental Results for gmiML-CFS and Other Multi-Label Feature Selection Methods Using the BPMLL Classifier

Clearly, in Tables 4.35 - 4.39, gmiML-CFS obtained substantially better predictive accuracy (substantially lower mean rank) than NoFS, CFS-U and RFML in most cases. In Table 4.35, reporting results for the small datasets, gmiML-CFS obtained the best overall average rank, 1.9; while NoFS, BR, CFS-U and RFML obtained larger overall average ranks (3.7, 2.3, 2.3 and 3.4, respectively). Moreover, gmiML-CFS outperformed all other approaches according to two predictive accuracy measures: Avg-Pre and R-Loss; and it was also the best method (jointly with BR) according to the OneError measure.

In Table 4.36, reporting results for the large datasets with feature space size equal to 100, BR obtained the best overall average rank (1.3); while gmiML-CFS obtained overall average rank 2.7 and outperformed NoFS, CFS-U and RFML with average rank 4.0, 3.3 and 3.7, respectively.

In Table 4.37, where the feature space size is equal to 200, again BR obtained the best result, with overall average rank 1.3. In addition, gmiML-CFS outperformed NoFS, CFS-U and RFML, with overall average rank = 2.6. Also, gmiML-CFS obtained better ranks than those three approaches on all five predictive accuracy measures in this table.

In Tables 4.38 and 4.39 again BR was the winner, with overall mean rank 1.3. In addition, in these two tables, gmiML-CFS outperformed NoFS, BR and RFML on all ten datasets, with overall average rank = 2.4 and 2.3, respectively. Moreover, gmiML-CFS obtained better ranks than those three approaches on all five predictive accuracy measures when the feature space size is equal to 300 or 400

Table 4.35: Values of five multi-label predictive accuracy measures for the best ML-CFS and other feature selection method using BPMLL as the classifier - small datasets

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
CAL500	NoFS	0.503	2.0	129.093	2.0	0.260	3.0	0.136	4.0	0.181	4.0	3.0
	BR(BPNN)	0.490	5.0	130.950	5.0	0.139	1.0	0.120	2.0	0.184	5.0	3.6
	CFS-U	0.507	1.0	127.480	1.0	0.257	2.0	0.152	5.0	0.176	1.0	2.0
	RFML	0.499	4.0	129.450	3.0	0.301	5.0	0.116	1.0	0.180	3.0	3.2
	gmi-ML-CFS	0.501	2.0	129.622	3.0	0.280	2.0	0.122	1.0	0.179	1.0	1.8
Scene	NoFS	0.501	4.0	1.994	5.0	0.301	5.0	0.747	4.0	0.380	5.0	4.6
	BR(BPNN)	0.540	3.0	1.960	4.0	0.235	2.0	0.667	3.0	0.370	4.0	3.2
	CFS-U	0.564	1.0	1.737	1.0	0.255	3.0	0.649	1.0	0.323	1.0	1.4
	RFML	0.553	2.0	1.885	2.0	0.228	1.0	0.655	2.0	0.357	2.0	1.8
	gmi-ML-CFS	0.497	3.0	1.918	3.0	0.272	3.0	0.789	3.0	0.361	3.0	3.0
Emotions	NoFS	0.791	4.0	1.889	3.0	0.214	2.0	0.309	4.0	0.168	3.0	3.2
	BR(BPNN)	0.797	1.0	1.876	2.0	0.209	1.0	0.282	1.0	0.159	1.0	1.2
	CFS-U	0.795	3.0	1.857	1.0	0.216	3.0	0.307	3.0	0.165	2.0	2.4
	RFML	0.779	5.0	1.995	5.0	0.238	5.0	0.331	5.0	0.193	5.0	5.0
	gmi-ML-CFS	0.795	1.0	1.930	1.0	0.220	2.0	0.294	3.0	0.172	2.0	1.8
Yeast	NoFS	0.738	5.0	6.619	3.0	0.227	2.0	0.263	5.0	0.190	5.0	4.0
	BR(BPNN)	0.757	1.0	6.364	1.0	0.198	1.0	0.242	2.0	0.171	1.0	1.2
	CFS-U	0.742	3.0	6.601	2.0	0.229	4.0	0.256	4.0	0.187	4.0	3.4
	RFML	0.741	4.0	6.674	5.0	0.232	5.0	0.237	1.0	0.187	3.0	3.6
	gmi-ML-CFS	0.742	1.0	6.644	1.0	0.228	1.0	0.245	1.0	0.183	1.0	1.0
MEAN	NoFS		3.8		3.3		3.0		4.3		4.3	3.7
	BR(BPNN)		2.5		3.0		1.3		2.0		2.8	2.3
	CFS-U		2.0		1.3		3.0		3.3		2.0	2.3
	RFML		3.8		3.8		4.0		2.3		3.3	3.4
	gmi-ML-CFS		1.8		2.0		2.0		2.0		1.8	1.9

(Tables 4.38 and 4.39).

Table 4.40 reports the summary of results in terms of the overall average rank and the number of selected features obtained by gmiML-CFS and multi-label feature selection approaches when using BPMLL as classifier. BR obtains the best average rank regarding accuracy but the largest used feature subset when compared with others (RFML, CFS-U and gmiML-CFS). For example; in CAL500, BR uses a feature subset about five times larger than the one selected by gmiML-CFS (68 and 12.90 features, respectively); and when we set the feature space size to 400 BR obtains the best predictive accuracy with a used feature subset about 7 times larger than the one selected by gmiML-CFS (400 and 57 features, respectively). The difference between the average ranks of BR and gmiML-CFS was small for the small datasets (where the original number of features is below 300) except on the CAL500 dataset, where the difference is 1.8. Note that the difference in the

Table 4.36: Values of five multi-label predictive accuracy measures for the best ML-CFS and other feature selection method using BPMLL as the classifier - feature space size = 100

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Business	NoFS	0.853	2.0	2.730	2.0	0.043	4.0	0.139	4.0	0.049	3.0	3.0
	BR(BPNN)	0.874	1.0	2.369	1.0	0.028	1.0	0.124	1.0	0.043	1.0	1.0
	CFS-U	0.850	4.0	2.818	4.0	0.043	5.0	0.139	4.0	0.050	4.0	4.2
	RFML	0.849	5.0	2.826	5.0	0.042	2.0	0.139	4.0	0.050	5.0	4.2
	gmi-ML-CFS	0.853	3.0	2.751	3.0	0.042	3.0	0.139	2.0	0.048	2.0	2.6
Art	NoFS	0.431	5.0	6.054	5.0	0.238	5.0	0.752	3.5	0.179	5.0	4.7
	BR(BPNN)	0.529	1.0	5.306	1.0	0.059	1.0	0.592	1.0	0.146	1.0	1.0
	CFS-U	0.438	2.0	5.909	2.0	0.218	4.0	0.752	3.5	0.172	2.0	2.7
	RFML	0.436	3.5	6.006	4.0	0.190	3.0	0.752	3.5	0.175	3.5	3.5
	gmi-ML-CFS	0.436	3.5	6.006	3.0	0.184	2.0	0.752	3.5	0.175	3.5	3.1
Education	NoFS	0.476	5.0	4.697	5.0	0.146	4.0	0.681	4.0	0.111	5.0	4.6
	BR(BPNN)	0.543	1.0	3.938	1.0	0.041	1.0	0.602	1.0	0.093	1.0	1.0
	CFS-U	0.476	4.0	4.683	4.0	0.133	2.0	0.681	4.0	0.111	4.0	3.6
	RFML	0.478	3.0	4.576	3.0	0.146	5.0	0.681	4.0	0.109	3.0	3.6
	gmi-ML-CFS	0.480	2.0	4.532	2.0	0.134	3.0	0.679	2.0	0.107	2.0	2.2
Recreation	NoFS	0.376	5.0	5.648	5.0	0.350	5.0	0.804	3.5	0.224	5.0	4.7
	BR(BPNN)	0.536	1.0	4.333	1.0	0.058	1.0	0.595	1.0	0.157	1.0	1.0
	CFS-U	0.381	2.0	5.447	3.0	0.224	4.0	0.804	3.5	0.215	2.0	2.9
	RFML	0.377	4.0	5.461	4.0	0.191	3.0	0.805	5.0	0.219	4.0	4.0
	gmi-ML-CFS	0.380	3.0	5.402	2.0	0.190	2.0	0.802	2.0	0.215	3.0	2.4
Health	NoFS	0.612	4.0	4.040	5.0	0.130	4.0	0.489	4.0	0.079	5.0	4.4
	BR(BPNN)	0.631	1.0	3.784	1.0	0.049	1.0	0.476	1.0	0.075	1.0	1.0
	CFS-U	0.611	5.0	4.024	4.0	0.130	5.0	0.489	5.0	0.078	4.0	4.6
	RFML	0.618	3.0	4.002	3.0	0.114	3.0	0.488	3.0	0.078	3.0	3.0
	gmi-ML-CFS	0.623	2.0	3.927	2.0	0.108	2.0	0.481	2.0	0.076	2.0	2.0
Enter.ment	NoFS	0.495	4.0	3.547	4.0	0.233	5.0	0.715	5.0	0.137	4.0	4.4
	BR(BPNN)	0.597	1.0	3.135	1.0	0.056	1.0	0.537	1.0	0.116	1.0	1.0
	CFS-U	0.523	3.0	3.460	3.0	0.162	3.0	0.662	3.0	0.132	3.0	3.0
	RFML	0.473	5.0	3.932	5.0	0.184	4.0	0.715	4.0	0.153	5.0	4.6
	gmi-ML-CFS	0.529	2.0	3.460	2.0	0.149	2.0	0.649	2.0	0.132	2.0	2.0
Computer	NoFS	0.598	4.0	4.876	3.0	0.093	5.0	0.475	2.0	0.103	4.0	3.6
	BR(BPNN)	0.630	1.0	4.289	1.0	0.040	1.0	0.443	1.0	0.091	1.0	1.0
	CFS-U	0.594	5.0	4.876	4.0	0.089	4.0	0.475	4.0	0.104	5.0	4.4
	RFML	0.598	3.0	4.893	5.0	0.073	2.0	0.475	4.0	0.102	3.0	3.4
	gmi-ML-CFS	0.599	2.0	4.867	2.0	0.084	3.0	0.475	4.0	0.101	2.0	2.6
Science	NoFS	0.393	5.0	7.873	5.0	0.212	5.0	0.758	3.5	0.158	5.0	4.7
	BR(BPNN)	0.456	1.0	6.852	1.0	0.035	1.0	0.676	1.0	0.134	1.0	1.0
	CFS-U	0.397	3.0	7.682	2.0	0.160	4.0	0.758	3.5	0.155	2.0	2.9
	RFML	0.397	2.0	7.747	4.0	0.123	2.0	0.758	3.5	0.155	3.0	2.9
	gmi-ML-CFS	0.397	4.0	7.747	3.0	0.124	3.0	0.758	3.5	0.156	4.0	3.5
Enron	NoFS	0.576	2.0	13.913	5.0	0.091	5.0	0.409	5.0	0.100	5.0	4.4
	BR(BPNN)	0.584	1.0	13.380	2.0	0.058	1.0	0.396	1.0	0.097	2.0	1.4
	CFS-U	0.573	3.0	13.811	4.0	0.090	3.0	0.397	2.0	0.100	4.0	3.2
	RFML	0.569	4.0	13.465	3.0	0.091	4.0	0.402	3.0	0.099	3.0	3.4
	gmi-ML-CFS	0.567	5.0	13.207	1.0	0.089	2.0	0.403	4.0	0.097	1.0	2.6
Medical	NoFS	0.796	2.0	2.606	2.0	0.018	1.0	0.271	2.0	0.042	2.0	1.8
	BR(BPNN)	0.717	3.0	3.614	4.0	0.019	3.0	0.374	3.0	0.062	3.0	3.2
	CFS-U	0.805	1.0	2.296	1.0	0.018	2.0	0.265	1.0	0.035	1.0	1.2
	RFML	0.582	4.0	3.808	5.0	0.052	5.0	0.588	4.0	0.070	5.0	4.6
	gmi-ML-CFS	0.557	5.0	3.604	3.0	0.050	4.0	0.655	5.0	0.066	4.0	4.2
MEAN	NoFS		3.8		4.1		4.3		3.7		4.3	4.0
	BR(BPNN)		1.2		1.4		1.2		1.2		1.3	1.3
	CFS-U		3.2		3.1		3.6		3.4		3.1	3.3
	RFML		3.7		4.1		3.3		3.8		3.8	3.7
	gmi-ML-CFS		3.2		2.3		2.6		3.0		2.6	2.7

Table 4.37: Values of five multi-label predictive accuracy measures for the best ML-CFS and other feature selection method using BPMLL as the classifier - feature space size = 200

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Business	NoFS	0.853	3.0	2.728	3.0	0.041	2.0	0.139	4.0	0.049	4.0	3.2
	BR(BPNN)	0.876	1.0	2.299	1.0	0.028	1.0	0.124	1.0	0.041	1.0	1.0
	CFS-U	0.855	2.0	2.705	2.0	0.041	3.0	0.139	4.0	0.048	2.0	2.6
	RFML	0.847	5.0	2.873	5.0	0.044	5.0	0.139	4.0	0.051	5.0	4.8
	gmi-ML-CFS	0.853	4.0	2.751	4.0	0.041	4.0	0.139	2.0	0.049	3.0	3.4
Art	NoFS	0.404	5.0	7.565	5.0	0.548	5.0	0.752	3.5	0.230	5.0	4.7
	BR(BPNN)	0.519	1.0	5.319	1.0	0.059	1.0	0.605	1.0	0.147	1.0	1.0
	CFS-U	0.428	4.0	6.184	4.0	0.287	4.0	0.752	3.5	0.183	4.0	3.9
	RFML	0.436	3.0	6.016	3.0	0.193	3.0	0.752	3.5	0.176	3.0	3.1
	ML-CFS	0.437	2.0	5.985	2.0	0.185	2.0	0.752	3.5	0.175	2.0	2.3
Education	NoFS	0.469	5.0	5.298	5.0	0.261	5.0	0.681	3.5	0.122	5.0	4.7
	BR(BPNN)	0.544	1.0	3.895	1.0	0.041	1.0	0.602	1.0	0.092	1.0	1.0
	CFS-U	0.476	4.0	4.687	4.0	0.154	4.0	0.681	3.5	0.111	4.0	3.9
	RFML	0.477	3.0	4.631	3.0	0.128	2.0	0.681	3.5	0.110	3.0	2.9
	gmi-ML-CFS	0.480	2.0	4.477	2.0	0.142	3.0	0.681	3.5	0.107	2.0	2.5
Recreation	NoFS	0.346	5.0	6.917	5.0	0.548	5.0	0.802	2.0	0.278	5.0	4.4
	BR(BPNN)	0.553	1.0	4.321	1.0	0.056	1.0	0.570	1.0	0.158	1.0	1.0
	CFS-U	0.370	4.0	5.939	4.0	0.386	4.0	0.805	4.5	0.237	4.0	4.1
	RFML	0.376	3.0	5.591	3.0	0.207	3.0	0.805	4.5	0.222	3.0	3.3
	gmi-ML-CFS	0.379	2.0	5.530	2.0	0.206	2.0	0.803	3.0	0.219	2.0	2.2
Health	NoFS	0.606	5.0	4.148	5.0	0.158	5.0	0.489	4.0	0.082	5.0	4.8
	BR(BPNN)	0.673	1.0	3.453	1.0	0.044	1.0	0.412	1.0	0.065	1.0	1.0
	CFS-U	0.609	4.0	4.098	4.0	0.152	4.0	0.489	4.0	0.080	4.0	4.0
	RFML	0.616	3.0	4.012	3.0	0.103	2.0	0.488	2.0	0.078	3.0	2.6
	gmi-ML-CFS	0.617	2.0	3.976	2.0	0.113	3.0	0.489	4.0	0.077	2.0	2.6
Enter.ment	NoFS	0.417	5.0	5.087	5.0	0.476	5.0	0.788	5.0	0.199	5.0	5.0
	BR(BPNN)	0.624	1.0	2.982	1.0	0.056	1.0	0.500	1.0	0.108	1.0	1.0
	CFS-U	0.480	3.0	3.799	3.0	0.266	4.0	0.715	3.5	0.149	3.0	3.3
	RFML	0.473	4.0	3.934	4.0	0.193	3.0	0.715	3.5	0.153	4.0	3.7
	gmi-ML-CFS	0.506	2.0	3.533	2.0	0.172	2.0	0.688	2.0	0.135	2.0	2.0
Computer	NoFS	0.582	4.0	5.111	5.0	0.169	5.0	0.475	3.5	0.111	5.0	4.5
	BR(BPNN)	0.647	1.0	4.125	1.0	0.038	1.0	0.424	1.0	0.087	1.0	1.0
	CFS-U	0.570	5.0	5.087	4.0	0.114	4.0	0.475	3.5	0.110	4.0	4.1
	RFML	0.598	3.0	4.904	3.0	0.072	2.0	0.475	3.5	0.103	3.0	2.9
	gmi-ML-CFS	0.601	2.0	4.810	2.0	0.084	3.0	0.475	3.5	0.101	2.0	2.5
Science	NoFS	0.382	5.0	9.138	5.0	0.478	5.0	0.758	3.5	0.188	5.0	4.7
	BR(BPNN)	0.476	1.0	6.617	1.0	0.034	1.0	0.654	1.0	0.129	1.0	1.0
	CFS-U	0.393	4.0	8.007	4.0	0.250	4.0	0.758	3.5	0.161	4.0	3.9
	RFML	0.398	2.0	7.689	2.0	0.131	3.0	0.758	3.5	0.154	2.0	2.5
	gmi-ML-CFS	0.396	3.0	7.811	3.0	0.129	2.0	0.758	3.5	0.157	3.0	2.9
Enron	NoFS	0.562	4.0	14.326	5.0	0.098	5.0	0.418	5.0	0.105	5.0	4.8
	BR(BPNN)	0.596	1.0	13.404	3.0	0.057	1.0	0.373	1.0	0.097	1.0	1.4
	CFS-U	0.572	3.0	13.969	4.0	0.092	4.0	0.409	4.0	0.102	4.0	3.8
	RFML	0.574	2.0	13.367	2.0	0.088	3.0	0.396	2.0	0.099	3.0	2.4
	gmi-ML-CFS	0.559	5.0	13.293	1.0	0.087	2.0	0.405	3.0	0.098	2.0	2.6
Medical	NoFS	0.759	2.0	2.588	4.0	0.019	3.0	0.353	4.0	0.041	3.0	3.2
	BR(BPNN)	0.745	4.0	3.557	5.0	0.019	2.0	0.321	2.0	0.060	5.0	3.6
	CFS-U	0.836	1.0	2.200	1.0	0.014	1.0	0.219	1.0	0.033	1.0	1.0
	RFML	0.698	5.0	2.534	3.0	0.030	5.0	0.460	5.0	0.043	4.0	4.4
	ML-CFS	0.758	3.0	2.475	2.0	0.026	4.0	0.341	3.0	0.040	2.0	2.8
MEAN	NoFS		4.3		4.7		4.5		3.8		4.7	4.4
	BR(BPNN)		1.3		1.6		1.1		1.1		1.4	1.3
	CFS-U		3.4		3.4		3.6		3.5		3.4	3.5
	RFML		3.3		3.1		3.1		3.5		3.3	3.3
	gmi-ML-CFS		2.7		2.2		2.7		3.1		2.2	2.6

Table 4.38: Values of five multi-label predictive accuracy measures for the best ML-CFS and other feature selection method using BPMLL as the classifier - feature space size = 300

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Business	NoFS	0.842	5.0	2.841	5.0	0.052	5.0	0.139	3.5	0.052	5.0	4.7
	BR(BPNN)	0.876	1.0	2.288	1.0	0.028	1.0	0.124	1.0	0.041	1.0	1.0
	CFS-U	0.853	2.0	2.757	3.0	0.038	3.0	0.139	3.5	0.049	4.0	3.1
	RFML	0.852	4.0	2.752	2.0	0.042	4.0	0.139	3.5	0.049	3.0	3.3
	gmi-ML-CFS	0.853	3.0	2.762	4.0	0.034	2.0	0.139	3.5	0.049	2.0	2.9
Art	NoFS	0.167	5.0	10.195	5.0	0.626	5.0	0.973	5.0	0.347	5.0	5.0
	BR(BPNN)	0.521	1.0	5.256	1.0	0.060	1.0	0.607	1.0	0.144	1.0	1.0
	CFS-U	0.421	4.0	7.123	4.0	0.519	4.0	0.752	3.0	0.212	4.0	3.8
	RFML	0.436	2.0	6.016	3.0	0.196	2.0	0.752	3.0	0.176	2.0	2.4
	gmi-ML-CFS	0.433	3.0	6.014	2.0	0.205	3.0	0.752	3.0	0.177	3.0	2.8
Education	NoFS	0.213	5.0	9.325	5.0	0.495	5.0	0.917	5.0	0.247	5.0	5.0
	BR(BPNN)	0.541	1.0	3.914	1.0	0.041	1.0	0.604	1.0	0.092	1.0	1.0
	CFS-U	0.472	4.0	4.983	4.0	0.169	4.0	0.681	3.5	0.117	4.0	3.9
	RFML	0.477	3.0	4.631	3.0	0.135	3.0	0.681	3.5	0.110	3.0	3.1
	gmi-ML-CFS	0.482	2.0	4.474	2.0	0.122	2.0	0.678	2.0	0.106	2.0	2.0
Recreation	NoFS	0.184	5.0	8.551	5.0	0.702	5.0	0.972	5.0	0.356	5.0	5.0
	BR(BPNN)	0.552	1.0	4.296	1.0	0.056	1.0	0.573	1.0	0.157	1.0	1.0
	CFS-U	0.370	4.0	6.133	4.0	0.481	4.0	0.805	3.5	0.242	4.0	3.9
	RFML	0.376	3.0	5.601	3.0	0.217	2.0	0.805	3.5	0.222	3.0	2.9
	gmi-ML-CFS	0.379	2.0	5.561	2.0	0.222	3.0	0.802	2.0	0.220	2.0	2.2
Health	NoFS	0.595	5.0	4.348	5.0	0.189	5.0	0.489	3.0	0.088	5.0	4.6
	BR(BPNN)	0.674	1.0	3.441	1.0	0.045	1.0	0.418	1.0	0.065	1.0	1.0
	CFS-U	0.606	4.0	4.254	4.0	0.181	4.0	0.489	3.0	0.084	4.0	3.8
	RFML	0.608	3.0	4.109	3.0	0.109	2.0	0.489	3.0	0.082	3.0	2.8
	gmi-ML-CFS	0.612	2.0	3.906	2.0	0.122	3.0	0.490	5.0	0.076	2.0	2.8
Enter.ment	NoFS	0.217	5.0	7.002	5.0	0.532	5.0	0.951	5.0	0.304	5.0	5.0
	BR(BPNN)	0.608	1.0	3.034	1.0	0.057	1.0	0.523	1.0	0.111	1.0	1.0
	CFS-U	0.471	4.0	4.054	4.0	0.314	4.0	0.715	3.5	0.157	4.0	3.9
	RFML	0.473	3.0	3.930	3.0	0.204	3.0	0.715	3.5	0.153	3.0	3.1
	gmi-ML-CFS	0.529	2.0	3.455	2.0	0.154	2.0	0.649	2.0	0.132	2.0	2.0
Computer	NoFS	0.235	5.0	8.556	5.0	0.475	5.0	0.971	5.0	0.211	5.0	5.0
	BR(BPNN)	0.651	1.0	4.086	1.0	0.037	1.0	0.423	1.0	0.086	1.0	1.0
	CFS-U	0.588	4.0	5.205	4.0	0.207	4.0	0.475	2.0	0.111	4.0	3.6
	RFML	0.595	2.5	5.006	3.0	0.083	2.0	0.475	3.5	0.106	2.5	2.7
	gmi-ML-CFS	0.595	2.5	5.003	2.0	0.083	3.0	0.475	3.5	0.106	2.5	2.7
Science	NoFS	0.153	5.0	12.225	5.0	0.546	5.0	0.981	5.0	0.268	5.0	5.0
	BR(BPNN)	0.475	1.0	6.611	1.0	0.034	1.0	0.660	1.0	0.130	1.0	1.0
	CFS-U	0.388	4.0	8.727	4.0	0.453	4.0	0.758	3.0	0.177	4.0	3.8
	RFML	0.399	2.0	7.664	2.0	0.132	3.0	0.758	3.0	0.154	2.0	2.4
	gmi-ML-CFS	0.396	3.0	7.815	3.0	0.129	2.0	0.758	3.0	0.157	3.0	2.8
Enron	NoFS	0.583	1.0	14.041	4.0	0.106	5.0	0.425	3.0	0.101	3.0	3.2
	BR(BPNN)	0.567	3.0	13.629	2.0	0.059	1.0	0.404	2.0	0.100	2.0	2.0
	CFS-U	0.569	2.0	14.361	5.0	0.090	3.0	0.427	4.0	0.104	5.0	3.8
	RFML	0.552	5.0	13.768	3.0	0.093	4.0	0.432	5.0	0.103	4.0	4.2
	gmi-ML-CFS	0.567	4.0	13.217	1.0	0.089	2.0	0.400	1.0	0.097	1.0	1.8
Medical	NoFS	0.215	5.0	9.014	5.0	0.198	5.0	0.940	5.0	0.181	5.0	5.0
	BR(BPNN)	0.738	4.0	3.578	4.0	0.019	2.0	0.336	3.0	0.060	4.0	3.4
	CFS-U	0.847	1.0	2.078	1.0	0.014	1.0	0.205	1.0	0.031	1.0	1.0
	RFML	0.753	3.0	2.810	3.0	0.023	4.0	0.339	4.0	0.045	3.0	3.4
	gmi-ML-CFS	0.796	2.0	2.344	2.0	0.021	3.0	0.282	2.0	0.036	2.0	2.2
MEAN	NoFS		4.6		4.9		5.0		4.5		4.8	4.8
	BR(kNN)		1.5		1.4		1.1		1.3		1.4	1.3
	CFS-U		3.3		3.7		3.5		3.0		3.8	3.5
	RFML		3.1		2.8		2.9		3.6		2.9	3.0
	gmi-ML-CFS		2.6		2.2		2.5		2.7		2.2	2.4

Table 4.39: Values of five multi-label predictive accuracy measures for the best ML-CFS and other feature selection method using BPMLL as the classifier - feature space size = 400

Dataset	Methods	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Business	NoFS	0.579	5.0	4.664	5.0	0.349	5.0	0.475	5.0	0.100	5.0	5.0
	BR(BPNN)	0.881	1.0	2.258	1.0	0.028	1.0	0.119	1.0	0.039	1.0	1.0
	CFS-U	0.856	3.0	2.646	2.0	0.041	3.0	0.139	3.5	0.047	2.0	2.7
	RFML	0.855	4.0	2.675	4.0	0.042	4.0	0.139	2.0	0.047	4.0	3.6
	gmi-ML-CFS	0.857	2.0	2.668	3.0	0.037	2.0	0.139	3.5	0.047	3.0	2.7
Art	NoFS	0.151	5.0	11.617	5.0	0.460	4.0	0.984	5.0	0.397	5.0	4.8
	BR(BPNN)	0.509	1.0	5.342	1.0	0.060	1.0	0.631	1.0	0.147	1.0	1.0
	CFS-U	0.337	4.0	8.150	4.0	0.544	5.0	0.843	4.0	0.257	4.0	4.2
	RFML	0.436	2.0	6.000	3.0	0.199	2.0	0.752	2.5	0.175	2.0	2.3
	gmi-ML-CFS	0.435	3.0	5.997	2.0	0.202	3.0	0.752	2.5	0.176	3.0	2.7
Education	NoFS	0.121	5.0	11.883	5.0	0.497	5.0	0.987	5.0	0.342	5.0	5.0
	BR(BPNN)	0.535	1.0	3.950	1.0	0.042	1.0	0.611	1.0	0.093	1.0	1.0
	CFS-U	0.470	4.0	5.194	4.0	0.233	4.0	0.681	3.0	0.120	4.0	3.8
	RFML	0.475	3.0	4.710	3.0	0.133	3.0	0.681	3.0	0.112	3.0	3.0
	gmi-ML-CFS	0.482	2.0	4.531	2.0	0.111	2.0	0.681	3.0	0.107	2.0	2.2
Recreation	NoFS	0.159	5.0	10.782	5.0	0.567	5.0	0.975	5.0	0.447	5.0	5.0
	BR(BPNN)	0.552	1.0	4.238	1.0	0.057	1.0	0.576	1.0	0.155	1.0	1.0
	CFS-U	0.334	4.0	6.674	4.0	0.547	4.0	0.840	4.0	0.270	4.0	4.0
	RFML	0.375	3.0	5.693	3.0	0.259	2.0	0.805	2.5	0.225	3.0	2.7
	gmi-ML-CFS	0.376	2.0	5.650	2.0	0.266	3.0	0.805	2.5	0.223	2.0	2.3
Health	NoFS	0.308	5.0	7.135	5.0	0.404	5.0	0.883	5.0	0.173	5.0	5.0
	BR(BPNN)	0.692	1.0	3.303	1.0	0.043	1.0	0.395	1.0	0.061	1.0	1.0
	CFS-U	0.587	4.0	4.688	4.0	0.217	4.0	0.489	3.5	0.096	4.0	3.9
	RFML	0.608	3.0	4.045	3.0	0.121	3.0	0.489	3.5	0.080	3.0	3.1
	gmi-ML-CFS	0.621	2.0	3.821	2.0	0.120	2.0	0.487	2.0	0.073	2.0	2.0
Enter.ment	NoFS	0.202	5.0	7.131	5.0	0.576	5.0	0.974	5.0	0.310	5.0	5.0
	BR(BPNN)	0.617	1.0	2.997	1.0	0.057	1.0	0.510	1.0	0.110	1.0	1.0
	CFS-U	0.461	4.0	4.371	4.0	0.367	4.0	0.715	3.0	0.169	4.0	3.8
	RFML	0.473	3.0	3.950	3.0	0.246	3.0	0.715	4.0	0.154	3.0	3.2
	gmi-ML-CFS	0.505	2.0	3.590	2.0	0.194	2.0	0.688	2.0	0.139	2.0	2.0
Computer	NoFS	0.135	5.0	11.156	5.0	0.574	5.0	0.983	5.0	0.301	5.0	5.0
	BR(BPNN)	0.655	1.0	4.030	1.0	0.037	1.0	0.418	1.0	0.084	1.0	1.0
	CFS-U	0.363	4.0	7.035	4.0	0.451	4.0	0.848	4.0	0.158	4.0	4.0
	RFML	0.595	3.0	5.006	3.0	0.083	2.0	0.475	2.5	0.106	3.0	2.7
	gmi-ML-CFS	0.599	2.0	4.888	2.0	0.084	3.0	0.475	2.5	0.103	2.0	2.3
Science	NoFS	0.128	5.0	14.598	5.0	0.592	5.0	0.980	5.0	0.329	5.0	5.0
	BR(BPNN)	0.462	1.0	6.680	1.0	0.035	1.0	0.671	1.0	0.132	1.0	1.0
	CFS-U	0.269	4.0	10.384	4.0	0.489	4.0	0.893	4.0	0.219	4.0	4.0
	RFML	0.398	2.0	7.708	2.0	0.144	2.0	0.758	2.5	0.156	2.0	2.1
	gmi-ML-CFS	0.397	3.0	7.733	3.0	0.146	3.0	0.758	2.5	0.156	3.0	2.9
Enron	NoFS	0.553	4.0	14.663	4.0	0.124	5.0	0.431	4.0	0.111	4.0	4.2
	BR(BPNN)	0.583	1.0	13.397	2.0	0.056	1.0	0.382	1.0	0.098	2.0	1.4
	CFS-U	0.552	5.0	14.828	5.0	0.096	4.0	0.435	5.0	0.112	5.0	4.8
	RFML	0.564	3.0	13.407	3.0	0.092	3.0	0.403	3.0	0.100	3.0	3.0
	gmi-ML-CFS	0.565	2.0	13.340	1.0	0.087	2.0	0.394	2.0	0.098	1.0	1.6
Medical	NoFS	0.154	5.0	14.135	5.0	0.325	5.0	0.940	5.0	0.292	5.0	5.0
	BR(BPNN)	0.728	4.0	3.716	4.0	0.020	2.0	0.349	4.0	0.063	4.0	3.6
	CFS-U	0.788	2.0	2.196	1.0	0.017	1.0	0.318	3.0	0.033	1.0	1.6
	RFML	0.780	3.0	2.514	3.0	0.022	4.0	0.299	2.0	0.041	3.0	3.0
	gmi-ML-CFS	0.804	1.0	2.395	2.0	0.020	3.0	0.267	1.0	0.037	2.0	1.8
MEAN	NoFS		4.9		4.9		4.9		4.9		4.9	4.9
	BR(BPNN)		1.3		1.4		1.1		1.3		1.4	1.3
	CFS-U		3.8		3.6		3.7		3.7		3.6	3.7
	RFML		2.9		3.0		2.8		2.8		2.9	2.9
	gmi-ML-CFS		2.1		2.1		2.5		2.4		2.2	2.3

Table 4.40: Summary of results in terms of average ranking (AR) and the number of selected features (Sel.F) of gmiML-CFS and other multi-label feature selection methods using BPMLL as the classifier

Datasets and	**NoFS**		**BR(BPNN)**		**CFS-U**		**RFML**		**gmi-ML-CFS**	
feature space size	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**
Emotion	3.20	72.00	1.20	72.00	2.40	52.00	5.00	10.00	1.80	10.00
CAL500	3.00	68.00	3.60	68.00	2.00	51.00	3.20	12.90	1.80	12.90
Scene	4.60	294.00	3.20	294.00	1.40	234.00	1.80	36.00	3.00	36.00
Yeast	4.00	103.00	1.20	103.00	3.40	74.00	3.60	22.00	1.00	22.00
100	4.03	100.00	1.26	100.00	3.27	73.90	3.72	22.40	2.72	22.40
200	4.40	200.00	1.30	200.00	3.46	128.40	3.26	34.30	2.58	34.30
300	4.75	300.00	1.34	300.00	3.46	174.80	3.03	44.10	2.42	44.10
400	4.90	400.00	1.30	400.00	3.68	214.40	2.87	57.00	2.25	57.00

average rank of BR and gmiML-CFS is particularly small (just 0.2) on the Scene and Yeast datasets. For the large datasets, the difference is decreasing from 1.46 to 1.28, 1.08 and 0.95 when the feature space size is equal to 100, 200, 300 and 400 respectively.

Figure 4.2 shows the overall average ranking (AR) for gmiML-CFS and the other multi-label feature selection methods plotted against the average size of selected features across all datasets and feature space sizes, when using BPMLL as the classifier. Again, clearly, gmi-ML-CFS obtains a very good tread-off between predictive accuracy (minimizing average ranking) and minimizing the number of selected features (analogous to the situation in Figure 4.1).

In general, gmiML-CFS selected the smallest feature subset while obtaining the second best predictive accuracy out of five different multi-label feature selection approaches.

Table 4.41 shows the overall average rank of five multi-label feature selection methods for each dataset (averaged across the 4 feature space sizes). The first value in each cell is the actual average rank, whilst the value between brackets is the "rank of the average rank". This later value was used in the Friedman and Holm's test. Using the Friedman's test we confidently conclude that there are sig-

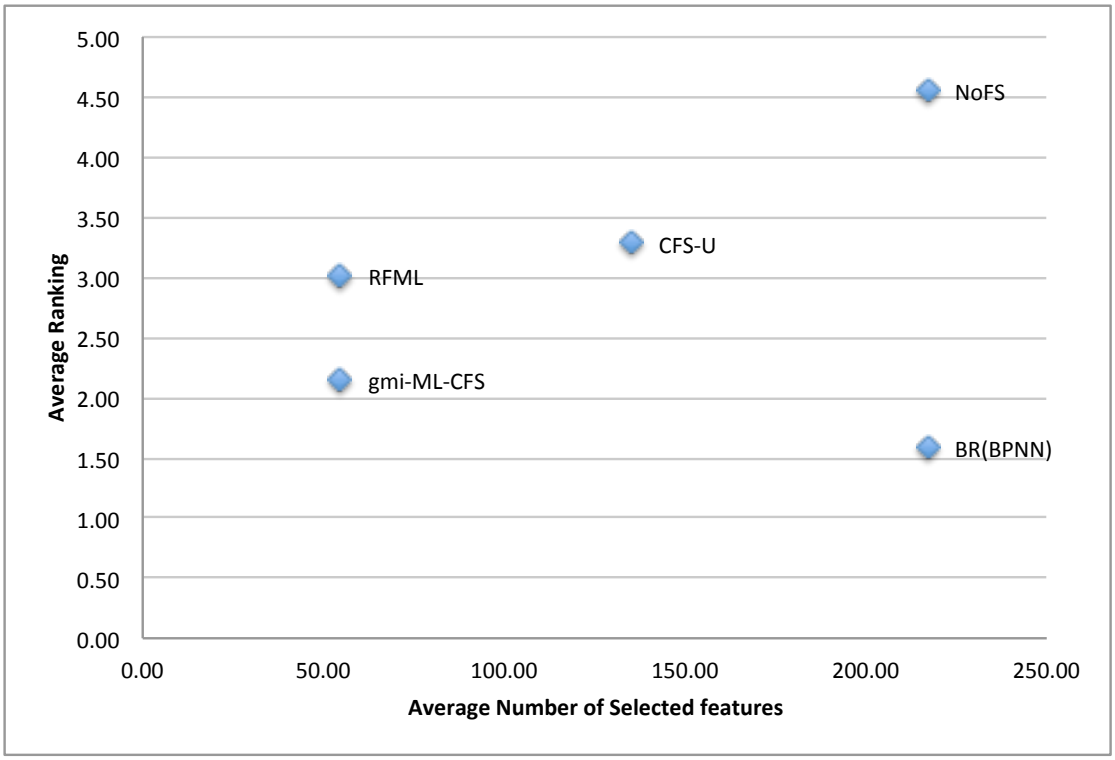


Figure 4.2: Overall average ranking (AR) for gmiML-CFS and the other multi-label feature selection methods plotted against the average size of selected features across all datasets and feature space sizes, when using BPMLL as the classifier

nificant differences among the 5 algorithms on 14 evaluation datasets at the 0.05 significance level for a two tailed test. Then, the Holm's posthoc test was applied on these data using gmiML-CFS as the control method. There is a significant difference between gmiML-CFS and NoFS at the 0.05 significant level (p value = 0.00012) but there are no significance differences between gmiML-CFS and the other 4 methods at the same level of significance.

4.6 Conclusion

This Chapter presented four versions of the Multi-Label Correlation Based Feature Selection (ML-CFS) method, based on hill climbing search. The first version of ML-CFS [57] extends the single-label CFS method to the more complex multi-

Table 4.41: Summary of overall average ranking (AR) gmiML-CFS and other multi-label feature selection methods across four feature space sizes using BPMLL as the classifier

Dataset	**Overall Average Rank (AR) across 4 feature space sizes**				
	NoFS	**BR(BPNN)**	**CFS-U**	**RFML**	**gmi-ML-CFS**
CAL500	3(3)	3.6(5)	2(2)	3.2(4)	1.8(1)
Scene	4.6(5)	3.2(4)	1.4(1)	1.8(2)	3(3)
Emotions	3.2(4)	1.2(1)	2.4(3)	5(5)	1.8(2)
Yeast	4(5)	1.2(2)	3.4(3)	3.6(4)	1(1)
Enron	4.2(4)	1.4(1)	4.8(5)	3(3)	1.6(2)
Medical	5(5)	3.6(4)	1.6(1)	3(3)	1.8(2)
Business	5(5)	1(1)	2.7(2)	3.6(4)	2.7(2)
Art	4.8(5)	1(1)	4.2(4)	2.3(2)	2.7(3)
Education	5(5)	1(1)	3.8(4)	3(3)	2.2(2)
Recreation	5(5)	1(1)	4(4)	2.7(3)	2.3(2)
Health	5(5)	1(1)	3.9(4)	3.1(3)	2(2)
Ent.ment	5(5)	1(1)	3.8(4)	3.2(3)	2(2)
Computer	5(5)	1(1)	4(4)	2.7(3)	2.3(2)
Science	5(5)	1(1)	4(4)	2.1(2)	2.9(3)
Average	4.56(4.71)	1.59(1.79)	3.29(3.21)	3.02(3.14)	2.15(2.07)

label classification scenario by computing the correlation between a feature and each of the multiple class labels. Then other three extensions of ML-CFS were proposed [58] namely; (1) ML-CFS with the Absolute Value of Correlation Coefficient (ML-CFSabs), (2) the ML-CFS version where class labels with greater mutual information (with respect to other labels) are assigned greater weight when computing feature-label correlations (gmiML-CFS); and (3) the ML-CFS version where class labels with greater mutual information are assigned smaller weights (smiML-CFS). Importantly, both gmiML-CFS and smiML-CFS also use the absolute value of correlation coefficient, since ML-CFSabs obtained in general substantially better results than the first version of ML-CFS.

We have run experiments with those four versions of ML-CFS and other multi-label feature selection methods to compare the predictive accuracy associated with their selected features when those features are used by two well-known multi-label classification algorithms: ML-kNN and BPMLL. From the experimental results reported in this Chapter, gmiML-CFS clearly outperforms ML-CFS, ML-CFSabs and smiML-CFS in general. Moreover, when comparing gmiML-CFS with other

multi-label feature selection methods, gmiML-CFS still shows a good predictive performance (it obtained the second best predictive accuracy out of five feature selection approaches) when using both classifiers. In addition, gmiML-CFS selects substantially smaller feature subsets than other methods which obtained the best predictive accuracy with both classifiers.

Chapter 5

Multi-Label Correlation-Based Feature Selection Methods that Exploit Biological Knowledge

In chapter 4, we proposed several versions of the Multi-Label Correlation-Based Feature Selection method (ML-CFS) and applied it to 14 multi-label datasets from a number of different application domains. In this Chapter we present extended versions of ML-CFS that exploit cancer-related information, in order to select a better set of genes (features) for cancer-related microarray datasets. This Chapter is organized as follows. Section 5.1 describes the general information about KEGG pathway. Section 5.2 describes three different versions of ML-CFS using KEGG pathway information. Section 5.3 describes the multi-label microarray datasets used in our experiments and Section 5.4 describes the experimental methodology. Section 5.5 reports experimental results and Section 5.6 presents this Chapter's conclusion.

5.1 A Feature Subset Evaluation Function for Exploiting Biological Knowledge

Recall that the original ML-CFS method evaluates the quality of a candidate feature subset by using a merit function, which rewards features that are highly correlated with the class attributes and have a low degree of redundancy with respect to other features. Hence, the merit function was designed to be independent from the application domain. Hence, in the context of the microarray datasets analyzed in this Chapter (datasets described in Section 5.3), the merit function has the limitation that it does not incorporate any biological knowledge about cancer-related genes. To improve the predictive accuracy and the potential for biological interpretation, int the context of cancer-related microarray datasets, we propose to extend the ML-CFS method with an evaluation function that uses some biological knowledge about cancer-related pathways.

Intuitively, the use of such biological knowledge would allow the ML-CFS method's search to focus on genes which are already known to be cancer-related, which could help to improve the predictive performance associated with the ML-CFS method or help to select genes whose role in cancer-related drug resistance or sensitivity is more likely to be meaningful to biologists.

More precisely, we use knowledge about cancer-related KEGG pathways, which is a well-known type of biological pathway, as part of the function that evaluates a candidate feature subset. [61, 84, 85].

A KEGG pathway is a set of genes or proteins and their interactions, broadly represented in the form of a graph. Each node typically represents a gene or protein, and an edge represents a type of interaction between genes or proteins. Some edges denote that a gene activates another, other edges denote that a gene or pro-

tein inhibits the activity of another, etc.

Moreover, KEGG pathways cover a wide range of organisms and are easy to use because each pathway is stored in well-known formats such as XML format files, text files and so on. KEGG pathways are widely used in literature [7, 41, 63].

Note that we utilize only 16 cancer-related KEGG pathways, which were selected based on current knowledge about the biology of cancer. The selection was made by Prof. Michaelis (School of BioSciences at University of Kent), an expert in cancer biology. Our experiments aim to select genes which are relevant for predicting drug sensitivity/resistance in cancer patients. So, it would not be effective to employ all pathways in the KEGG database. The selected 16 cancer-related KEGG pathways are:

- DNA replication
- Base excision repair
- Nucleotide excision repair
- Mismatch repair
- Homologous recombination
- Non-homologous end-joining
- Fanconi anemia pathway
- ABC transporters
- Wnt signaling pathway
- Notch signaling pathway
- Hedgehog signaling pathway
- Cell cycle
- Apoptosis
- p53 signaling pathway
- Pathways in cancer
- Transcriptional misregulation in cancer

Detailed information about these cancer-related pathways is provided on the KEGG website (http://www.genome.jp/kegg/). We assume that if some genes are related with cancer-related drug resistance/sensitivity, they are likely to occur in some of the above cancer-related pathways.

In order to quantify the strength of the relationship between the genes in a candidate feature subset and the aforementioned cancer-related pathways, we propose to compute "the Average Relative Frequency of Pathways per gene" (AvgRFP):

$$AvgRFP_{FSS_i} = \frac{\sum_{f=1}^{k} RFP_f}{k} \tag{5.1}$$

where the average is computed over all the k features selected to be included in the i-th candidate feature subset (FSS_i), as shown in Equation (5.1).

For each selected feature f in FSS_i, the relative frequency of pathways for f, denoted by RFP_f, is the number of cancer-related KEGG pathways in which the gene corresponding to f occurs divided by the number of user-specified pathways (16 in our case). Each RFP_f has a value in [0..1], so $AvgRFP_{FSS_i}$ also has a value in [0..1]. Hence, the $AvgRFP$ term rewards feature subsets where most genes in the subset are involved in several cancer-related pathways, and penalizes feature subsets where most genes do not occur in any cancer-related pathway.

5.2 Three extensions of Multi-Label Correlation-Based Feature Selection (ML-CFS) using KEGG Pathway Information

In this Section we propose three extensions to the original ML-CFS method, which exploit cancer-related knowledge. Two of these extensions use Equation (5.1),

whilst the third extension consists of using as input features only genes occurring in the selected KEGG pathways.

5.2.1 ML-CFS using a Weighted Formula to Combine the Merit Function and KEGG Pathway Information

In this approach, the evaluation function of the i-th *FSS* is defined by the following weighted formula:

$$EvaluationFunction = \alpha \times Merit_{FSS_i} + (1 - \alpha) \times AvgRFP_{FSS_i} \quad (5.2)$$

where α is a weight in [0..1] which is a user-defined parameter, whilst $Merit_{FSS_i}$ and $AvgRFP_{FSS_i}$ were discussed earlier.

The advantage of this approach is its simplicity: it computes the value of the merit of a candidate feature subset and its *AvgRFP* value separately (representing two different perspectives, one statistical and another biological, respectively). More precisely, the merit function evaluates candidate feature subsets using the concept of statistical correlation; while *AvgRFP* evaluates candidate feature subsets in terms of how often the genes in a feature subset occur in cancer-related KEGG pathways. An important point of our experiments is that we use $\alpha \geq 0.5$ i.e, the weight α assigned to the merit function ($Merit_{FSS_i}$) is greater than or equal to the weight $(1 - \alpha)$ assigned to *AvgRFP*. This is because we consider the predictive accuracy (evaluated by the merit function) as the primary evaluation criterion of a feature subset, while *AvgRFP* is a secondary (but still important to users) criterion supporting the discovery of biologically relevant features. There is no point in discovering biologically relevant features with low accuracy.

5.2.2 ML-CFS Embedding KEGG Pathway Information into the Merit Function

We also tried to embed the value of $AvgRFP$ into the merit function in order to avoid the need to specify user-defined weights (α) in our evaluation function. In this approach, the formula to calculate the average value of the correlation between all features in a feature subset F and all the labels in class label set L is different from the formula in the original ML-CFS. The new formula is as follows:

$$\overline{r_{FL}} = \frac{\sum_{f=1}^{|F|} \left| r_{f\overline{L}} \right| \times RFP_f}{\sum_{f=1}^{|F|} RFP_f} \tag{5.3}$$

The idea behind this formula is that we want to reward the feature-label correlation values in proportion to the strength of the association between the genes in a feature subset and the cancer-related KEGG pathways (as measured by the RFP term), while the average correlation between pairs of features in a feature subset (to detect redundancy) is computed in the same way as in the original ML-CFS algorithm.

The effect of using this formula with the hill climbing search used by ML-CFS is that the algorithm will select only genes which occur in some KEGG pathway in the first iteration of hill climbing search. This is because in the first iteration of the search each candidate feature subset contains just one feature (gene), and if that gene does not occur in any KEGG pathway the value of ($\overline{r_{FL}}$) is equal to zero because $RFP_f = 0$. In that case the value of the merit function is equal to zero because in the first iteration the average correlation between feature pairs in the feature subset ($\overline{r_{FF}}$) is ignored (there is no feature pair in the feature subset), so that only the correlation between features and labels ($\overline{r_{FL}}$) is considered.

After the first iteration of the hill climbing search, the candidate feature subsets will have at least one gene which occurs in at least one cancer-related KEGG

pathway and the correlation between features and labels ($\overline{r_{FL}}$) is taken into account. Therefore, a selected feature subset returned by ML-CFS will have at least one gene occurring in a cancer-related KEGG pathway; and the rest of the genes selected by ML-CFS's hill climbing search are expected not only to be highly correlated with class labels but also to have little redundancy with the other selected genes.

5.2.3 ML-CFS Using as Input Only Genes Occurring in the Selected KEGG Pathways

When using the approach of embedding KEGG pathway information into the merit function, there is a chance that the ML-CFS method selects a feature subset which has only one gene occurring in some cancer-related pathways and the rest of the selected genes are not occurring in any cancer related pathway at all. Note that, in our datasets, only 3.13 % of the genes (690 out of 22,060 genes) occur in some cancer-related KEGG pathway, and most of those genes have an *RFP* value lower than 0.15. Hence, we decided to do experiments with another approach which selects only genes which occur in cancer-related KEGG pathways. The idea behind this approach is to investigate what will happen if we force our feature selection method (ML-CFS) to select a feature subset from a feature space containing only the genes (features) that occur in some cancer-related pathway. Hence, in this approach we remove all genes which do not occur in any cancer-related pathway from the feature space. After that we give all the remaining genes (i.e. all the genes occurring in some cancer-related KEGG pathway) as input to the ML-CFS method.

Table 5.1: Main Characteristics of the Datasets used in the experiments

Dataset Symbol	Dataset Name	Dataset Description					
		Instances	Features	Labels	Label Cardinality	Label Density	Distinct Labels
M1	Nutlin.Rita	24	22060	3	0.869	0.434	4
M2	Cis.Car.Oxy	24	22058	3	1.304	0.434	4

5.3 Datasets Used in the Experiments

In our experiments, we have analysed two multi-label microarray gene expression datasets (Table 5.1). Unlike the other datasets analyzed in the previous Chapter, the two multi-label microarray datasets are not publically available; they were prepared for data mining by the author of this thesis, using data provided by Prof. Michaelis, School of Bioscience, University of Kent. Both these datasets were obtained from the resistant cancer cell line (RCCL) collection [22]. The first one (referred to as dataset M1) consists of 28,536 features (genes), 24 instances (cell lines) and 2 class attributes. More precisely, each feature represents the (real-valued) expression level of a different gene, for each cell line (instance) in the dataset. The two class attributes stand for two drugs which are used to treat neuroblastoma (a type of cancer), namely: 'Nutlin-3', which can take two class labels (sensitive and resistant), and 'RITA', which can take three class labels (sensitive, resistant and highly resistant) for each cell line. Hence, the goal of the multi-label classification algorithm is to produce a classification model that, given the values of the features (gene expression levels) for a cell line, predicts whether that cell line would be sensitive or resistant to the drug Nutlin-3, and predicts whether that cell line would be sensitive, resistant or highly resistant to the drug RITA.

In order to prepare dataset M1 for the application of a multi-label algorithm, first we decompose the two class attributes into three binary class labels. The first binary class label (L_1) indicates whether a cell line (an instance) is sensitive or resistant to drug Nutlin-3. The situation is more complicated in the case of the class attribute for the RITA drug, which can take 3 values, since conventional multi-label algorithms can cope only with binary class labels. Hence, we decom-

posed the 3 class values for RITA into two binary attributes: L_2 takes the value yes or no to indicate whether or not a cell line is sensitive to the RITA drug; whilst L_3 takes the value yes or no to indicate whether or not a cell line is highly resistant to RITA. Hence, at most one of labels L_2 and L_3 can take the value yes for a given cell line. If both L_2 and L_3 take the value no for a cell line, this means the cell line is resistant to the drug RITA. Also, if L_1, L_2 and L_3 take the value no for a cell line, this means the cell line is sensitive to Nutlin-3 and resistant to RITA. Note that the fact that several cell lines have this pattern of three labels with value no leads to an average value of label cardinality smaller than 1, since label cardinality is computed by counting the number of yes values in labels.

The second multi-label microarray dataset – referred to as M2 – also has 28,536 features (genes) and 24 instances (cell lines), but it has 3 binary class attributes (different drugs used to treat neuroblastoma), namely: Cisplatin, Carboplatin and Oxaliplating.

Moreover, in both dataset M1 and M2, we remove genes with unknown names because we aimed at selecting genes whose relevance to drug resistance/sensitivity can be interpreted by biologists. After removing unknown genes, the number of features (genes) that remained in dataset M1 is 22060, and 22,058 genes (features) remained in dataset M2 (each dataset had about 22.7% of genes with unknown names).

5.4 Experimental Methodology

The experiments reported in this Chapter are devided into five parts, as follows. First, we ran an experiment for comparing the two different versions of ML-CFS: (1) the first version of ML-CFS (described in Section 4.1); and (2) the ML-CFS method using the absolute value of correlation coefficient (ML-CFSabs), which

Table 5.2: Five different versions of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information

Methods	α	$1-\alpha$
ML-CFSk55	0.5	0.5
ML-CFSk64	0.6	0.4
ML-CFSk73	0.7	0.3
ML-CFSk82	0.8	0.2
ML-CFSk91	0.9	0.1

was described in Section 4.2.1.

Second, we ran an experiment for comparing 5 different parameter (α) settings of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information, as described in Section 5.2. The pre-defined weights (α) and ($1-\alpha$) used in Equation (5.2) are shown in Table 5.2.

Third, we compare the best version of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information against other two versions of ML-CFS: (1) ML-CFSabs; and (2) gmiML-CFS, the ML-CFS version where class labels with greater MI (Mutual Information) are assigned greater weights (described in Section 4.2.2). The idea of this experiment is to evaluate what extent the use of mutual information and KEGG pathway information improve over ML-CFSabs ability to select a high quality feature subset. It is important to mention that gmiML-CFS also uses the absolute value of the correlation coefficient (like ML-CFSabs).

Fourth, we compare the best version of ML-CFS according to the result of the previous experiment against other two ML-CFS versions using KEGG pathway information: (1) ML-CFS with embedded KEGG pathway Information to the Merit Function (described in Section 5.2.2); and (2) ML-CFS selecting only genes that occur in KEGG pathways (described in Section 5.2.3).

Fifth, we compare the best version of our ML-CFS method in the previous experiment against Relief for Multi-Label feature selection (RFML), and the proposed Correlation-Based Feature Selection with the union operator (CFS-U) . These are the same baseline approaches used in the previous Chapter, and the details of each approach are described in Section 4.5.1.

The results of these five experiments are reported in Sections 5.5.1 through 5.5.5, respectively. In each of these five experiments, in order to evaluate the predictive performance of the different versions of ML-CFS, the feature subset selected by each ML-CFS version was given to two different types of multi-label classification algorithm, namely the Multi-Label k-Nearest Neighbour (ML-kNN) classification algorithm proposed in [124] and the Back-Propagation Multi-Label Learning (BPMLL) classification algorithm [123]. These two algorithms were run using their default parameters, which were mentioned in their corresponding paper. After that, the predictive accuracy of each classification model was measured, for each ML-CFS version, on the test set, containing data instances, which were not included in the training set, therefore measuring the generalization ability of the classification model. For the two microarray datasets (M1 and M2) we used the well-known leave one out cross-validation procedure [116].

Like in Chapter 4, we measure predictive accuracy using five different accuracy measures, namely: Hamming-loss, Ranking-loss, One-error, Coverage and Average Precision [113], as reviewed in Chapter 2.

5.5 Experimental Results

5.5.1 Experimental Results for the First version of ML-CFS and ML-CFS with the Absolute Value of Correlation Coefficient

Tables 5.3 and 5.4 show the predictive performance of the first version of ML-CFS (denoted simply by ML-CFS) and the ML-CFS with absolute value of correlation coefficient (ML-CFSabs) on two microarray datasets (described in Section 5.3). In these datasets, ML-CFS was applied to the full set of features; which was feasible despite the very large number of features, because the number of instances is very small.

In Tables 5.3 and 5.4 the numbers in each column titled "R" denote the ranks achieved by each method according to the accuracy measure in the corresponding left column. The ranks vary in the range from 1 (best) to 2 (worst). The tables also report, in the last column, the average rank (AR) of each method across all five predictive accuracy measures, for each dataset.

The last two rows of each table show the mean rank for each method across two datasets, In those last two rows, the mean value of each accuracy measure is not reported because that mean value would not be vary meaningful, since the different datasets have different degrees of difficult for a classification algorithm, so that different accuracies across datasets cannot be fairly compared, as mentioned In Chapter 4. On the other hand, it is fair to compare the rank of the ML-CFS versions across the two datasets, so the mean ranks are reported. Finally the last column of the last two rows shows the average ranks over the five predictive accuracy measures and over the two datasets.

Table 5.3: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using ML-kNN as the classifier

Dataset	Method.	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
M1	ML-CFS	0.868	2.0	0.333	2.0	0.375	2.0	0.417	2.0	0.208	2.0	2.0
	ML-CFSabs	0.974	1.0	0.167	1.0	0.292	1.0	0.250	1.0	0.042	1.0	1.0
M2	ML-CFS	0.618	2.0	0.750	2.0	0.333	2.0	0.500	2.0	0.125	2.0	2.0
	ML-CFSabs	0.640	1.0	0.708	1.0	0.319	1.0	0.458	1.0	0.083	1.0	1.0
Mean	ML-CFS		2.00		2.00		2.00		2.00		2.00	2.0
	ML-CFSabs		1.00		1.00		1.00		1.00		1.00	1.0

Table 5.4: Values of five multi-label predictive accuracy measures for the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using BPMLL as the classifier

Dataset	Method.	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
M1	ML-CFS	0.842	2.0	0.375	2.0	0.421	2.0	0.458	2.0	0.250	2.0	2.0
	ML-CFSabs	0.974	1.0	0.167	1.0	0.175	1.0	0.250	1.0	0.042	1.0	1.0
M2	ML-CFS	0.994	2.0	0.633	2.0	0.189	2.0	0.383	2.0	0.008	2.0	2.0
	ML-CFSabs	1.000	1.0	0.625	1.0	0.175	1.0	0.375	1.0	0.000	1.0	1.0
Mean	ML-CFS		2.00		2.00		2.00		2.00		2.00	2.0
	ML-CFSabs		1.00		1.00		1.00		1.00		1.00	1.0

Clearly, in Tables 5.3 and 5.4, ML-CFSabs obtained substantially better predictive accuracy (substantially lower mean rank) than ML-CFS for each of the five accuracy measures in both tables. ML-CFSabs outperforms ML-CFS on the two datasets with overall average rank = 1.0. Since, ML-CFSabs obtains the better rank for all five predictive accuracy measures with both classifiers.

Table 5.5 shows the summary of results reported in Tables 5.3 and 5.4, by reporting the average rank and the average number of features selected by ML-CFS and ML-CFSabs on the two microarray datasets, where all features were available to ML-CFS and ML-CFSabs. ML-CFSabs obtains the best average rank (1.0); while ML-CFS obtains the worst rank (2.0). In terms of the number of selected features, ML-CFSabs selected a larger number of features in all cases, across the two datasets and using both ML-kNN and BPMLL classifiers (as shown in the column titled "S.F").

Table 5.5: Summary of average ranking (Avg.R) and the number of selected features (S.F) obtained by the first version of ML-CFS and ML-CFS with absolute value of correlation coefficient using ML-kNN and BPMLL as classifiers

MLkNN				
Dataset	**ML-CFS**		**ML-CFSabs**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**
M1	2.96	2.0	8.79	1.0
M2	2.96	2.0	2.96	1.0
Mean	2.96	2.0	5.87	1.0
BPMLL				
Dataset	**ML-CFS**		**ML-CFSabs**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**
M1	2.96	2.0	8.79	1.0
M2	2.96	2.0	2.96	1.0
Mean	2.96	2.0	5.87	1.0

5.5.2 Experimental Results for Five Versions of ML-CFS Using a Weighted Formula to Combine the Merit Function and KEGG Pathway Information

Tables 5.6 and 5.7 show the predictive performance of five versions of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information on the two microarray datasets using ML-kNN and BPMLL, respectively. Recall that, ML-CFSk55, ML-CFSk64, ML-CFSk73, ML-CFSk82 and ML-CFSk91 stand for the ML-CFSabs using Equation 5.2 to combine the merit function and KEGG pathway information, where the number after "k" refers to the different weight settings (α and $1-\alpha$) as mentioned in Table 5.2.

Clearly, in Table 5.6, ML-CFSk91 (where $\alpha = 0.9$ and $1-\alpha = 0.1$) obtained substantially better predictive accuracy (substantially lower mean rank) than the other ML-CFS versions for each of the five accuracy measures. As can be seen in Equation(5.2), ML-CFSk91 assigns the largest weight to the merit function and the smallest weight to the AvgRFP term (exploiting biological knowledge about cancer-relates pathways). However, there is no general correlation between larger

value of α and better ranks, since the second best overall rank was obtained by ML-CFSk55, which has the lowest value of α (0.5) among the 5 versions of ML-CFS in Table 5.2.

ML-CFSk91 outperforms other versions of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information on the two datasets with overall average rank = 1.2. Also, ML-CFSabs obtains the better rank for all five predictive accuracy measures on ML-kNN classifiers.

In Tables 5.7, ML-CFSk91 outperforms other versions of ML-CFS using a weighted formula to combine the merit function and KEGG pathway Information on two microarray datasets with overall average rank = 1.4. Also, ML-CFSk91 obtains the better rank for all five predictive accuracy measures with the BPMLL classifier. However, again there is no clear correlation between the value of α and the corresponding overall average rank. The average ranks alternate decrease and increase as the value of α is varied from 0.5 to 0.9.

Table 5.8 shows the summary of results reported in Tables 5.6, and 5.7, by reporting the average rank and the average number of features selected by five different versions of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information on the two microarray datasets, where all features were available. When using ML-kNN as the classifier, ML-CFSk91 obtains the best average rank (1.2); while ML-CFSk55 wins the second place with the average rank equal to 2.25 and the worst method is ML-CFSk64 which obtains the worst rank (4.05) with ML-kNN. When using the BPMLL classifier, ML-CFSk91 obtains the best average rank (1.4); while ML-CFSk55 wins the second place with the average rank equal to 2.25 and the worst method is ML-CFSk55 which obtains the worst rank (3.95).

In terms of the mean of selected features across the two datasets M1 and

Table 5.6: Values of five multi-label predictive accuracy measures for five versions of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information using ML-kNN as the classifier

Dataset	ML-CFS versions	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
M1	ML-CFSk55	0.974	3.5	0.167	3.5	0.396	2.5	0.250	3.5	0.042	3.5	3.3
	ML-CFSk64	0.974	3.5	0.167	3.5	0.417	4.5	0.250	3.5	0.042	3.5	3.7
	ML-CFSk73	0.974	3.5	0.167	3.5	0.417	4.5	0.250	3.5	0.042	3.5	3.7
	ML-CFSk82	0.974	3.5	0.167	3.5	0.396	2.5	0.250	3.5	0.042	3.5	3.3
	ML=CFSk91	1.000	1.5	0.125	1.0	0.271	2.0	0.208	1.0	0.000	1.5	1.4
M2	ML-CFSk55	1.000	1.0	0.625	1.0	0.194	2.0	0.375	1.0	0.000	1.0	1.2
	ML-CFSk64	0.917	4.5	0.750	4.5	0.347	4.0	0.500	4.5	0.125	4.5	4.4
	ML-CFSk73	0.972	2.0	0.667	2.0	0.097	1.0	0.417	2.0	0.042	2.0	1.8
	ML-CFSk82	0.944	3.0	0.708	3.0	0.347	5.0	0.458	3.0	0.083	3.0	3.4
	ML-CFSk91	0.917	1.0	0.750	1.0	0.306	1.0	0.500	1.0	0.125	1.0	1.0
MEAN	ML-CFSk55		2.25		2.25		2.25		2.25		2.25	2.25
	ML-CFSk64		4.00		4.00		4.25		4.00		4.00	4.05
	ML-CFSk73		2.75		2.75		2.75		2.75		2.75	2.75
	ML-CFSk82		3.25		3.25		3.75		3.25		3.25	3.35
	ML-CFSk91		1.25		1.00		1.50		1.00		1.25	1.20

Table 5.7: Values of five multi-label predictive accuracy measures for five versions of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information using BPMLL as the classifier

Dataset	ML-CFS version	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
M1	ML-CFSk55	0.974	5.0	0.167	5.0	0.171	2.5	0.250	5.0	0.042	5.0	4.5
	ML-CFSk64	0.984	2.0	0.150	2.0	0.171	2.5	0.233	2.0	0.025	2.0	2.1
	ML-CFSk73	0.984	2.0	0.150	2.0	0.167	1.0	0.233	2.0	0.025	2.0	1.8
	ML-CFSk82	0.979	4.0	0.158	4.0	0.188	4.0	0.242	4.0	0.033	4.0	4.0
	ML-CFSk91	0.984	3.0	0.150	1.0	0.188	1.0	0.233	1.0	0.025	3.0	1.8
M2	ML-CFSk55	1.000	3.0	0.625	3.0	0.253	5.0	0.375	3.0	0.000	3.0	3.4
	ML-CFSk64	1.000	3.0	0.625	3.0	0.236	2.0	0.375	3.0	0.000	3.0	2.8
	ML-CFSk73	1.000	3.0	0.625	3.0	0.244	4.0	0.375	3.0	0.000	3.0	3.2
	ML-CFSk82	1.000	3.0	0.625	3.0	0.242	3.0	0.375	3.0	0.000	3.0	3.0
	ML-CFSk91	1.000	1.0	0.625	1.0	0.197	1.0	0.375	1.0	0.000	1.0	1.0
MEAN	ML-CFSk55		4.00		4.00		3.75		4.00		4.00	3.95
	ML-CFSk64		2.50		2.50		2.25		2.50		2.50	2.45
	ML-CFSk73		2.50		2.50		2.50		2.50		2.50	2.50
	ML-CFSk82		3.50		3.50		3.50		3.50		3.50	3.50
	ML-CFSk91		2.00		1.00		1.00		1.00		2.00	1.40

M2, as shown in Table 5.8, ML-CFSk55 tends to select the smallest number of features (4.25) when compared with ML-CFSk64, ML-CFSk73, ML-CFSk82 and ML-CFSk91 (which on average select 4.75, 5.46, 6.38 and 7.67 features as shown in the column titled "S.F" for those 4 methods, respectively).

Table 5.8: Summary of average ranking (Avg.R) and the number of selected features (S.F.) for five versions of ML-CFS using a weighted formula to combine the merit function and KEGG pathway information using ML-kNN and BPMLL as classifiers

ML-kNN										
Dataset	**ML-CFSk55**		**ML-CFSk64**		**ML-CFSk73**		**ML-CFSk82**		**ML-CFSk91**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**
M1	2.96	3.30	3.00	3.70	3.38	3.70	5.67	3.30	9.67	1.40
M2	5.54	1.20	6.46	4.40	7.54	1.80	7.08	3.40	5.67	1.00
MEAN	4.25	2.25	4.73	4.05	5.46	2.75	6.38	3.35	7.67	1.20
BPMLL										
Dataset	**ML-CFSk55**		**ML-CFSk64**		**ML-CFSk73**		**ML-CFSk82**		**ML-CFSk91**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**
M1	2.96	4.50	3.00	2.10	3.38	1.80	5.67	4.00	9.67	1.80
M2	5.54	3.40	6.46	2.80	7.54	3.20	7.08	3.00	5.67	1.00
MEAN	4.25	3.95	4.73	2.45	5.46	2.50	6.38	3.50	7.67	1.40

5.5.3 Experimental Results Comparing the Best Version of ML-CFS Using a Weighted Formula, ML-CFS with the Absolute Value of Correlation Coefficient and ML-CFS Using Mutual Information

Tables 5.9 and 5.10 show the predictive performance of ML-CFSk91 (the ML-CFS version which obtained the best results in the previous Section), ML-CFSabs; and gmiML-CFS using ML-kNN and BPMLL as classifier. Recall that ML-CFSk91 is the version of ML-CFS using the weighted formula (shown in Equation (5.2)) to combine the merit function and KEGG pathway Information (with $\alpha = 0.9$ and 1 - $\alpha = 0.1$); ML-CFSabs stands for ML-CFS with the Absolute value of correlation coefficient; and gmiML-CFS stands for the ML-CFS version where class labels with greater MI (Mutual Information) are assigned greater wrights. These tables report the predictive performance across the two microarray datasets using ML-kNN and BPMLL as classifiers, respectively.

Note that both gmiML-CFS and ML-CFSk91 also use the absolute value of the correlation coefficient (like ML-CFSabs). Hence, the experiment in this Section

allow us to observe the effect of using mutual information and exploiting biological knowledge on cancer-related pathways in a controlled manner. In addition, note that ML-CFSabs can be seen as a particular case of the use of Equation (5.2) where $\alpha = 1.0$ and $1 - \alpha = 0$.

In Table 5.9, ML-CFSk91 obtained substantially better predictive accuracy (substantially lower mean rank) than ML-CFSabs and gmiML-CFS. ML-CFSk91 outperforms other versions of ML-CFS on the two datasets with overall average rank = 1.8, while gmiML-CFS and ML-CFSabs obtain average rank 2.0 and 2.3 respectively when using ML-kNN classifier.

In Table 5.10, ML-CFSk91 outperforms ML-CFSabs and gmiML-CFS on the two microarray datasets with overall average rank = 1.8 while gmiML-CFS and ML-CFSabs both obtain average rank 2.1 respectively. Also, ML-CFSk91 obtains the better rank for all five predictive accuracy measures with the BPMLL classifier.

Table 5.11 shows the summary of the results reported in Tables 5.9 and 5.10, by reporting the average rank and the average number of features selected by gmiML-CFS, ML-CFSabs and ML-CFSk91. When using ML-kNN classifier, ML-CFSk91 obtains the best average rank (1.75); while gmiML-CFS takes the second place with the average rank equal to 2.00 and ML-CFSabs obtains the worst rank (2.25). When using BPMLL classifier, ML-CFSk91 again obtains the best average rank (1.8); while gmiML-CFS and ML-CFSabs obtain the same average rank (2.1).

In terms of the average number of selected features, as shown in Table 5.11, ML-CFSk91 tends to select the largest number of features (on average 7.67 over the two datasets), while the smallest number of selected features is obtained by ML-CFSabs, which selected on average 5.87 features over the two datasets (as shown in the column titled "S.F"). Overall, each of those three methods select less than 0.04% of all features in dataset.

Table 5.9: Values of five multi-label predictive accuracy measures for ML-CFSk91, ML-CFSabs and gmiML-CFS using ML-kNN as the classifier

Dataset	Method	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
M1	gmiML-CFS	0.974	2.5	0.167	2.5	0.271	1.5	0.250	2.5	0.042	2.5	2.3
	ML-CFSk91	1.000	1.0	0.125	1.0	0.271	1.5	0.208	1.0	0.000	1.0	1.1
	ML-CFSabs	0.974	2.5	0.167	2.5	0.292	3.0	0.250	2.5	0.042	2.5	2.6
M2	gmiML-CFS	0.944	1.0	0.708	1.5	0.347	3.0	0.458	1.5	0.083	1.5	1.7
	ML-CFSk91	0.917	2.0	0.750	3.0	0.306	1.0	0.500	3.0	0.125	3.0	2.4
	ML-CFSabs	0.640	3.0	0.708	1.5	0.320	2.0	0.458	1.5	0.083	1.5	1.9
Mean	gmiML-CFS		1.75		2.00		2.25		2.00		2.00	2.0
	ML-CFSk91		1.50		2.00		1.25		2.00		2.00	1.8
	ML-CFSabs		2.75		2.00		2.50		2.00		2.00	2.3

Table 5.10: Values of five multi-label predictive accuracy measures for ML-CFSk91, ML-CFSabs and gmiML-CFS using BPMLL as the classifier

Dataset	Method	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
M1	gmiML-CFS	0.974	2.5	0.167	2.5	0.179	2.0	0.250	2.5	0.042	2.5	2.4
	ML-CFSk91	0.984	1.0	0.150	1.0	0.188	3.0	0.233	1.0	0.025	1.0	1.4
	ML-CFSabs	0.974	2.5	0.167	2.5	0.175	1.0	0.250	2.5	0.042	2.5	2.2
M2	gmiML-CFS	1.000	2.0	0.625	2.0	0.172	1.0	0.375	2.0	0.000	2.0	1.8
	ML-CFSk91	1.000	2.0	0.625	2.0	0.197	3.0	0.375	2.0	0.000	2.0	2.2
	ML-CFSabs	1.000	2.0	0.625	2.0	0.175	2.0	0.375	2.0	0.000	2.0	2.0
Mean	gmiML-CFS		2.25		2.25		1.50		2.25		2.25	2.1
	ML-CFSk91		1.50		1.50		3.00		1.50		1.50	1.8
	ML-CFSabs		2.25		2.25		1.50		2.25		2.25	2.1

5.5.4 Experimental Results Comparing the Best Version of ML-CFS Using a Weighted Formula (ML-CFSk91), ML-CFS with Embedded KEGG pathway Information and ML-CFS Using Only Genes that Occur in KEGG Pathway

Tables 5.12 and 5.13 show the predictive performance of ML-CFSk91 (the ML-CFS version which obtained the best results in the previous Section), ML-CFSkemb; and ML-CFSflt using ML-kNN and BPMLL as classifiers. Recall that ML-CFSk91 is the version of ML-CFS using the weighted formula (shown in Equation (5.2)) to combine the merit function and KEGG pathway information (with $\alpha = 0.9$ anf $1 - \alpha = 0.1$); ML-CFSemb stands for ML-CFS with KEGG pathway Information embedded into the merit Function; and ML-CFSflt stands for ML-CFS with "fil-

Table 5.11: Summary of average ranking (Avg.R) and the number of selected features (S.F) for for ML-CFSk91, ML-CFSabs and gmiML-CFS using ML-kNN and BPMLL as classifiers

MLkNN						
Dataset	**gmiML-CFS**		**ML-CFSk91**		**ML-CFSabs**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**
M1	9.00	2.30	9.67	1.10	8.79	2.60
M2	5.00	1.70	5.67	2.40	2.96	1.90
Mean	7.00	2.00	7.67	1.75	5.87	2.25
BPMLL						
Dataset	**gmiML-CFS**		**ML-CFSk91**		**ML-CFSabs**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**
M1	9.00	2.4	9.67	1.4	8.79	2.2
M2	5.00	1.8	5.67	2.2	2.96	2.0
Mean	7.00	2.1	7.67	1.8	5.87	2.1

tered" genes, i.e, selecting only genes that occur in KEGG pathway. These tables report the predictive performance across the two microarray datasets using ML-kNN and BPMLL as classifiers, respectively.

Note that both ML-CFSemb and ML-CFSflt also use the absolute value of the correlation coefficient (like ML-CFSk91 and ML-CFSabs).

In Table 5.12, ML-CFSk91 obtained substantially better predictive accuracy (substantially lower mean rank) than ML-CFSemb and ML-CFSflt. ML-CFSk91 outperformed other versions of ML-CFS using KEGG pathway information on the two datasets with overall average rank = 1.3, while ML-CFSemb and ML-CFSflt obtained average rank 2.3 and 2.4 respectively when using ML-kNN classifier. Also, ML-CFSk91 obtained the better rank for four predictive accuracy measures (except H-Loss) when using the ML-kNN classifier.

In Table 5.13, ML-CFSk91 outperformed ML-CFSemb and ML-CFSflt on the two microarray datasets with overall average rank = 1.2; while ML-CFSemb and ML-CFSflt obtained average rank 2.9 and 2.0 respectively. Also, ML-CFSk91 ob-

Table 5.12: Values of five multi-label predictive accuracy measures for ML-CFSk91, ML-CFS with KEGG pathway information embedded into the Merit Function and ML-CFS selecting only genes that occur in KEGG pathway using ML-kNN as the classifier

Dataset	Method	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
M1	ML-CFSk91	1.000	1.0	0.125	1.0	0.271	1.0	0.208	1.0	0.000	1.0	1.0
	ML-CFSemb	0.947	2.0	0.208	2.0	0.396	2.0	0.292	2.0	0.083	2.0	2.0
	ML-CFSflt	0.921	3.0	0.250	3.0	0.438	3.0	0.333	3.0	0.125	3.0	3.0
M2	ML-CFSk91	0.917	1.5	0.750	1.5	0.306	2.0	0.500	1.5	0.125	1.5	1.6
	ML-CFSemb	0.856	3.0	0.875	3.0	0.292	1.0	0.542	3.0	0.167	3.0	2.6
	ML-CFSflt	0.917	1.5	0.750	1.5	0.347	3.0	0.500	1.5	0.125	1.5	1.8
Mean	ML-CFSk91		1.25		1.25		1.50		1.25		1.25	1.3
	ML-CFSemb		2.50		2.50		1.50		2.50		2.50	2.3
	ML-CFSflt		2.25		2.25		3.00		2.25		2.25	2.4

Table 5.13: Values of five multi-label predictive accuracy measures for ML-CFSk91, ML-CFS with KEGG pathway information embedded into the Merit Function and ML-CFS selecting only genes that occur in KEGG pathway using BPMLL as the classifier

Dataset	Method	Predictive Accuracy										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
M1	ML-CFSk91	0.984	1.0	0.150	1.0	0.1875	1.0	0.233	1.0	0.025	1.0	1.0
	ML-CFSemb	0.968	3.0	0.175	3.0	0.225	3.0	0.258	3.0	0.050	3.0	3.0
	ML-CFSflt	0.974	2.0	0.167	2.0	0.208	2.0	0.25	2.0	0.042	2.0	2.0
M2	ML-CFSk91	1.000	1.0	0.625	1.0	0.197	1.0	0.375	2.0	0.000	1.5	1.3
	ML-CFSemb	0.913	2.0	0.775	3.0	0.350	3.0	0.475	3.0	0.100	3.0	2.8
	ML-CFSflt	0.8	3.0	0.652	2.0	0.290	2.0	0.348	1.0	0	1.5	1.9
Mean	ML-CFSk91		1.00		1.00		1.00		1.50		1.25	1.2
	ML-CFSemb		2.50		3.00		3.00		3.00		3.00	2.9
	ML-CFSflt		2.50		2.00		2.00		1.50		1.75	2.0

tains the better rank for all five predictive accuracy measures with the BPMLL classifier.

Table 5.14 shows the summary of the results reported in Tables 5.12 and 5.13, by reporting the average rank and the average number of features selected by ML-CFSk91, ML-CFSemb and ML-CFSflt. When using the ML-kNN classifier, ML-CFSk91 obtains the best average rank (1.30); while ML-CFSemb takes the second place with the average rank equal to 2.30 and ML-CFSflt obtains the worst rank (2.40). When using the BPMLL classifier, ML-CFSk91 again obtains the best average rank (1.15); while ML-CFSflt and ML-CFSemb obtain the average rank 1.95 and 2.90, respectively.

Table 5.14: Summary of average ranking (Avg.R) and the number of selected features (S.F.) for for ML-CFSk91, ML-CFSemb and ML-CFSflt using ML-kNN and BPMLL as classifiers

MLkNN						
Dataset	**ML-CFSk91**		**ML-CFSemb**		**ML-CFSflt**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**
M1	9.67	1.00	17.08	2.00	7.42	3.00
M2	5.67	1.60	16.42	2.60	3.83	1.80
Mean	7.67	1.30	16.75	2.30	5.63	2.40
BPMLL						
Dataset	**ML-CFSk91**		**ML-CFSemb**		**ML-CFSflt**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**
M1	9.67	1.00	17.08	3.00	7.42	2.00
M2	5.67	1.30	16.42	2.80	3.83	1.90
Mean	7.67	1.15	16.75	2.90	5.63	1.95

In terms of the aaverage number of selected features, as shown in Table5.14,ML-CFSemb tends to select the largest number of features (on average 16.75 over the two datasets), while the smallest number of selected features is obtained by ML-CFSflt, which selected on average only 5.63 features (as shown in the column titled "S.F.").

5.5.5 Computational Results Comparing the Best Version of ML-CFS (ML-CFSk91) and Two Other Multi-Label Feature Selection Methods

In this Section we compare the best version of our ML-CFS using KEGG pathway information according to the results reported in previous Section, namely ML-CFSk91, with Relief for Multi-Label feature selection (RFML) and Correlation-Based Feature Selection with the union operator (CFS-U). The details of RFML and CFS-U were described in Section 4.5.1. Note that in this Section we report results separately for the experiments using ML-kNN and BPMLL as the classifier.

Table 5.15: Values of five multi-label predictive accuracy measures for ML-CFSk91 and other feature selection methods using ML-kNN as the classifier

Dataset	Method	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
M1	CFS-U	1.000	1.5	0.125	1.5	0.313	2.0	0.208	1.5	0.000	1.5	1.6
	RFML	0.921	3.0	0.250	3.0	0.604	3.0	0.333	3.0	0.125	3.0	3.0
	ML-CFSk91	1.000	1.5	0.125	1.5	0.271	1.0	0.208	1.5	0.000	1.5	1.4
M2	CFS-U	0.917	2.0	0.750	2.0	0.569	3.0	0.500	2.0	0.125	2.0	2.2
	RFML	0.917	2.0	0.750	2.0	0.403	2.0	0.500	2.0	0.125	2.0	2.0
	ML-CFSk91	0.917	2.0	0.750	2.0	0.306	1.0	0.500	2.0	0.125	2.0	1.8
Mean	CFS-U		1.8		1.8		2.5		1.8		1.8	1.9
	RFML		2.5		2.5		2.5		2.5		2.5	2.5
	ML-CFSk91		1.8		1.8		1.0		1.8		1.8	1.6

Table 5.16: Values of five multi-label predictive accuracy measures for ML-CFSk91 and other feature selection methods using BPMLL as the classifier

Dataset	Method.	Predictive Accuracy										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
M1	CFS-U	1.000	1.0	0.125	1.0	0.171	1.0	0.208	1.0	0.000	1.0	1.0
	RFML	0.974	3.0	0.167	3.0	0.213	3.0	0.250	3.0	0.042	3.0	3.0
	ML-CFSk91	0.984	2.0	0.150	2.0	0.188	2.0	0.233	2.0	0.025	2.0	2.0
M2	CFS-U	1.000	2.0	0.565	2.0	0.151	2.0	0.391	3.0	0.000	2.0	2.2
	RFML	1.000	2.0	0.625	1.0	0.140	1.0	0.375	1.5	0.000	2.0	1.5
	ML-CFSk91	1.000	2.0	0.625	3.0	0.197	3.0	0.375	1.5	0.000	2.0	2.3
Mean	CFS-U		1.5		1.0		1.5		2.0		1.5	1.5
	RFML		2.5		2.8		2.0		2.3		2.5	2.4
	ML-CFSk91		2.0		2.3		2.5		1.8		2.0	2.1

Clearly, in Tables 5.15, ML-CFSk91 obtained substantially better predictive accuracy (substantially lower overall average rank) than RFML and CFS-U across the two microarray datasets. ML-CFSk91 obtained 1.6 average rank while CFS-U and RFML obtained 1.9 and 2.5, respectively. Interestingly, ML-CFSk91 obtains the same predictive accuracy as CFS-U for four measures (Avg.Pre, Coverage, OneError and R-Loss), but ML-CFSk91 obtained substantially better Hamming Loss.

In Table 5.16, CFS-U obtained the best average rank (1.5), which was substantially better than the average rank obtained by ML-CFSk91 (2.1) and RFML (2.4).

Table 5.17 shows the summary of the results reported in Tables 5.15 and 5.16, reporting the average rank and the average number of features selected by ML-CFSk91, RFML and CFS-U. Note that CFS-U tends to select by far the largest

Table 5.17: Summary of average ranking (Avg.R) and the number of selected features (S.F.) for for ML-CFSk91, RFML and CFS-U using ML-kNN and BPMLL as classifiers

MLkNN						
Dataset	**ML-CFSk91**		**CFS-U**		**RFML**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**
M1	9.67	1.40	1296.37	1.60	10.00	3.00
M2	5.67	1.80	332.79	2.20	6.00	2.00
Mean	7.67	1.60	814.58	1.90	8.00	2.50
BPMLL						
Dataset	**ML-CFSk91**		**CFS-U**		**RFML**	
	S.F.	**Avg.R**	**S.F.**	**Avg.R**	**S.F.**	**Avg.R**
M1	9.67	2.00	1296.37	1.00	10.00	3.00
M2	5.67	2.20	332.79	2.00	6.00	1.50
Mean	7.67	2.10	814.58	1.50	8.00	2.25

number of features on M1 and M2 (1296.37 and 332.79, respectively); while the average number of selected features was very small for ML-CFSk91 and RFML (7.67 and 8.00, respectively). When using the ML-kNN classifier, ML-CFSk91 obtained the best average rank across both datasets (1.6) with the smaller number of features comparing with CFS-U. On the other hand, when using the BPMLL, CFS-U obtained the best average rank (1.5) across both datasets.

5.6 Conclusion

This Chapter proposed three versions of the Multi-Label Correlation Based Feature Selection (ML-CFS) method exploiting cancer-related pathway information, based on hill climbing search. These three extensions of ML-CFS, introduced in [59], are as follows; (1) ML-CFS using a Weighted Formula to Combine the Merit Function and KEGG Pathway Information (ML-CFSk),(2) ML-CFS Embedding KEGG Pathway Information into the Merit Function (ML-CFSkemb); and (3) ML-CFS using as input only genes occurring in the selected KEGG pathway (ML-CFSfilt). Importantly, all those three versions of ML-CFS also use the absolute

value of correlation coefficient, since ML-CFSabs obtained in general substantially better results than the first version of ML-CFS in the experiments reported in Chapter 4.

Regarding ML-CFS using a Weighted Formula to Combine the Merit Function and KEGG Pathway Information, we run experiments using five different parameter settings. We have also run experiments to compare the best version of ML-CFSk (using the best parameter setting found in our experiments) with other two versions of ML-CFS method that exploit cancer-related pathway information. Then the best version of ML-CFS exploiting cancer-related pathway information was chosen to be compared with two other multi-label feature selection methods. We then measure the predictive accuracy associated with their selected features, when those features are used as input by two well-known multi-label classification algorithms: ML-kNN and BPMLL.

From the experimental results reported in this Chapter, ML-CFSk91 clearly outperformed ML-CFS, ML-CFSabs, ML-CFSemb, ML-CFS-flt and gmiML-CFS in general. Moreover, when comparing ML-CFSk91 with other multi-label feature selection methods, ML-CFSk91 obtained the best predictive accuracy out of three feature selection methods when using the ML-kNN classifier; and the second best accuracy when using the BPMLL classifier. In addition, ML-CFSk91 selects much smaller feature subsets than CFS-U, the method that obtained the best predictive accuracy with the BPMLL classifier.

Chapter 6

Multi-Label Correlation-Based Feature Selection Methods Based on Evolutionary Algorithms

6.1 Introduction

In chapter 4, we developed a Multi-Label Correlation-Based Feature Selection method (ML-CFS) and applied it to 14 multi-label datasets. In terms of the search strategy used to explore the space of candidate feature subsets, ML-CFS uses a simple greedy strategy. Recall that a greedy search algorithm uses a heuristic for making locally optimal choices at each stage with the hope of finding a global optimum. In the case of sequential forward greedy search (the search strategy we used), the algorithm starts with the empty set of candidate solutions, and creates new candidate feature subsets by adding one feature at a time into the current candidate feature subset. At each step, the feature to be added to the current candidate feature subset is the one with the best value of an evaluation function. This iterative process of selecting one feature at a time is performed until a termination criterion is satisfied (e.g. the quality of the current feature subset cannot be improved by adding any other feature).

This kind of greedy strategy, or hill-climbing search, performs only a local search in the space of candidate feature subsets, selecting just one feature at a time and so ignoring interactions between two or more features that could be added to the current candidate solution. The interaction between features is very important for the feature selection task. One feature may be useless by itself but potentially useful when we consider it together with other features. Overall, the hill-climbing search incrementally adds extra features to the current candidate feature subset if each of the extra features has a quality high enough to increase the value of the Merit function (Equation 6.1). In this scenario, the hill-climbing search strategy would conservatively reject the extra features if the requirement for improving merit value is not satisfied for a single extra feature being added, even though that feature might be useful when combined with extra features to be added later. This is a result of the hill-climbing search's limitation of adding just one feature at a time, which is not a very effective approach to cope with feature interaction [34, 35, 37].

Unlike the local greedy search strategy, Genetic Algorithms (GAs) are stochastic search methods inspired by the process of natural selection, based on Darwin's evolutionary theory [32]. A GA performs a more global search in the feature space than a greedy search, because a GA works with a population of candidate solutions spread across different regions of the search space. Moreover, genetic operators help the GA to explore a wider area of the search space [34], by comparison with local greedy search. As a result of their global search, GAs cope better with feature interaction and are less likely to get trapped into a local optimum in the search space, being more likely to find a global optimum.

The new Genetic Algorithm for Multi-Label Correlation-Based Feature Selection (GA-ML-CFS) proposed in this chapter extends our previous version of ML-CFS by replacing the simple greedy strategy by a more sophisticated GA as a

search method. The GA uses the genetic operators of crossover and mutation and a fitness-based selection method to explore the space of candidate feature subsets.

In the next two sections, we proposed two versions of a GA-based ML-CFS feature selection method: one version using a single-objective fitness function, described in Section 6.2, and another version based on lexicographic multi-objective optimization, described in Section 6.3. Section 6.4 mentions the datasets used in the experiments. Section 6.5 describes the experimental methodology used in this Chapter. Section 6.6 reports results for parameter optimization for single-objective GA and multi-objective-GA. Section 6.7 reports computational results comparing single-objective GAs and multi-objective GAs. Then, Section 6.8 reports the computational results comparing the best version of GA-ML-CFS (gmiGA-wrap) and other multi-label feature selection methods. A general discussion of the results will be presented in Section 6.9.

6.2 ML-CFS with a Single-Objective Genetic Algorithm (GA-ML-CFS)

The basic idea of GA-ML-CFS is that a GA is used as a search method for multi-label correlation-based feature selection. Hence, by comparison with the ML-CFS based on Hill-climbing search described in Chapter 4, GA-ML-CFS uses different search operators, but the same candidate solution representation and the same evaluation function. Algorithm 6.1 shows the overall pseudocode of GA-ML-CFS, where GenNum and MaxGen denote the current generation (iteration) number and the maximum number of generations, respectively. MaxGen is a user-specified parameter. First, all individuals are initialized and evaluated. Then, elitism is applied for protecting the best solutions in the current population from genetic operators, where ElitSize is a user-specified parameter. That is, the ElitSize best

individuals will be copied to the next generation, without any modification. After that, tournament selection is applied for finding good solutions that will undergo crossover and mutation. Next, population replacement is applied and then these process are repeated until the current generation number is equal to the maximum number of generations. The details of each process are described in Subsection 6.2.1 through Subsection 6.2.5.

Algorithm 6.1: Overall Pseudocode of GA-ML-CFS()

$CurrentPOP \leftarrow$ Initialize Population
Evaluate fitness of each individual in CurrentPOP
GenNum = 0
repeat
  $ElitPool \leftarrow$ ElitSize best individuals in CurrentPOP
  $MatchPool \leftarrow$ Result of Tournament Selection applied to CurrentPOP
  $ChildPool \leftarrow$ Result of Crossover applied to individuals in MatchPool
  $ChildPool \leftarrow$ Result of Mutation applied to individuals in ChildPool
  $CurrentPOP \leftarrow$ ElitePool $\cup$ ChildPool
  Evaluate Fitness of each individual in CurrentPOP
$GenNum \leftarrow GenNum + 1$
until $GenNum > MaxGen$

6.2.1 Individual (Candidate Solution) Representation and Population Initialization

GA-ML-CFS uses a bit string individual representation. Each candidate solution is encoded by a string of n bits, where n is the number of features in the dataset. The i-th bit with value '1' indicates that the i-th feature was selected, while the i-th bit with value '0' indicates that the i-th feature was not selected.

Algorithm 6.2: Population Initialization step in GA-ML-CFS()

$/ * PopulationInitialization * /$

$CurrentPOP = null$

for individual $i \leftarrow 1$ **to** p

do
- **for each** gene of individual i
  - **do**
    - Generate random number z in [0..1]
    - Set the gene value for individual i to $\begin{cases} 1, \textbf{if } z \leq initProb \\ 0, otherwise \end{cases}$
- Add individual i to CurrentPOP

Algorithm 6.2 shows the pseudocode of the Population Initialization step in GA-ML-CFS. The value of each bit in each individual depends on a random number z and the initiation probability (*initProb*) - a user-defined parameter. In our experiments we decided to create an initial population where the value of *initProb* varies across individuals, because this leads to a greater diversity of the number of genes selected by different individuals. If the value of z is smaller than or equal to the value of *initProb* then the value of this bit will be set to 1. Otherwise the value of this bit will be set to 0. After this step we will have p individuals in the individual pool. The elitism strategy is applied after the population initialization step. As mentioned earlier, we preserved the top ElitSize individuals and put them into the new individual pool. Those ElitSize individuals will be passed to next generation directly (without performing crossover and mutation on them).

6.2.2 Parent Selection

The next step of the GA is the parent selection. In this step we apply the well-known tournament selection method to select parent individuals before performing

crossover and mutation operators. Tournament selection [9, 32] is a method which runs the tournament many times (also called the number of tournament rounds) for selecting individuals from the current population (individual pool). Each tournament round selects one individual. In Algorithm 6.3, in each tournament round t individuals are randomly drawn from the individual pool and they compete with each other. The individual who has the best fitness value will win the tournament. Ties are broken at random. As shown in Algorithm 6.3, we set the number of tournament rounds (MaxRound) to be equal to the population size (PopSize) minus the elitist set size (ElitSize) value (the number of individuals preserved for the next generation), and add the tournament winner to the match pool.

Algorithm 6.3: PARENT SELECTION IN GA-ML-CFS()

$MatchPool = null$

$MaxRound \leftarrow$ the PopSize $-$ ElitSize

for $tourRound \leftarrow 1$ **to** $MaxRound$

do { Randomly drawn t individuals from CurrentPOP
Add tournament winner to MatchPool

6.2.3 Genetic Search Operators

GA-ML-CFS uses uniform crossover and bit-flip mutation [32]. Uniform crossover generates a string of L random variables taking values in [0, 1], where L is the number of genes - i.e. the number of features in the dataset. In each position of the individual's string of genes, if the value of the corresponding random variable is lower than or equal to a pre-defined number geneCrossProb (the probability of crossover per gene), the gene values in this position are swapped between the two parents, to create two children. The procedure of uniform crossover is shown in Algorithm 6.4. Recall that MaxRound is the number of tournament rounds (see

Algorithm 6.3), and so the number of crossovers performed in Algorithm 6.4 is half of MaxRound, since each crossover acts on two individuals selected by the tournament selection procedure.

Algorithm 6.4: Uniform Crossover in GA-ML-CFS()

$ChildPool = null$

for $CrossRound \leftarrow 1$ **to** $MaxRound/2$

do
- Randomly match two individuals from MatchPool and remove them from that pool
- **for each** gene of the parent individuals
  - **do**
    - Generate random number z in [0..1]
    - **if** $z \leq geneCrossProb$
      - **then** switch gene values between the two parents
- Add the two individuals to *ChildPool*

After the crossover step, all individuals have a chance of performing mutation. To implement the mutation operator, a random number will be generated for each gene in each individual, and then we compare each random number with a user-defined gene mutation probability (geneMutProb). A mutation will be performed – i.e. the bit will be flipped in a given gene of a given individual – if the random number generated for that gene is smaller than or equal to the user-defined mutation probability.

Algorithm 6.5: MUTATION IN GA-ML-CFS()

```
for each individual in ChildPool
do { for each gene of the individual
     do { Generate random number z in [0..1]
          if z ≤ geneMutProb
            then { invert the binary gene value
```

6.2.4 Population Replacement

The fourth step of the GA is the population replacement step. The best ElitSize (elitist set size) individuals from the current generation are preserved and copied into the next generation. In Algorithm 6.6, we prepare the individuals for the next generation by setting CurrentPOP to the union of ElitPool and ChildPool, where ElitPool is a set of individuals preserved from elitism process, and ChildPool is the set of individuals after applying crossover and mutation to the current parents. Finally, the new generation will go through the parent selection, crossover and mutation steps again and so on. The GA will terminate when a user-specified number of generations has been executed.

Algorithm 6.6: POPULATION REPLACEMENT IN GA-ML-CFS()

```
Set CurrentPOP= ElitPool ∪ ChildPool
for individual i ← 1 to |CurrentPOP|
  do Calculate the Fitness (Merit) of individual i
```

6.2.5 Fitness (Evaluation) Function

Each individual (feature subset) in the population was evaluated using Equation 6.1. Recall that the terms in the merit formula were modified to use the absolute (without sign) value of the correlation coefficient, as shown in Equations 6.2 and 6.3, which compute the average correlation between features and class labels and the average correlation between all feature pairs, respectively.

$$Merit = \frac{k\overline{r_{FL}}}{\sqrt{k + k(k-1)\overline{r_{FF}}}} \tag{6.1}$$

$$\overline{r_{FL}} = \frac{\sum_{f=1}^{|F|} \left| r_{f\overline{L}} \right|}{|F|} \tag{6.2}$$

$$\overline{r_{FF}} = \frac{\sum_{f_i=1, f_j=1, i \neq j}^{|F|} \left| r_{f_i f_j} \right|}{f_p} \tag{6.3}$$

$$r_{f\overline{L}} = \frac{\sum_{i=1}^{|L|} |r_{fL_i}|}{|L|} \tag{6.4}$$

6.2.6 Parameters of the Genetic Algorithm

The GA has several user-defined parameters, namely: individual size (n), population size (PopSize), the number of generations (MaxGen), the elitist set size (ElitSize), the tournament size (t), gene crossover probability (geneCrossProb) and gene mutation probability (geneMutProb).

These parameters are optimized using a set of datasets different from the set of datasets used to measure the predictive accuracy associated with the GA; as explained later.

6.2.7 Data Preprocessing for the Genetic Algorithm

For datasets with a very large number of features, a univariate filter approach was applied to select features before running GA-ML-CFS, similarly to the use of the univariate filter approach in the experiments with ML-CFS based on hill climbing in Chapter 4. Recall that the main objective of this stage is to remove features which have a low correlation with class labels before running the GA, in order to reduce the GA's search space. The average correlation between a feature and all labels is measured using Equation 6.4. We calculate the average correlation coefficient between each feature f_i and all class labels (considering one feature at a time). After calculating the average correlation over all labels for each feature, we rank all features according to their average correlation value and select the top n features in the sorted list to be the set of n features given as input to the GA. Note that our proposed approach is different from other GAs for feature selection mentioned in Chapter 2 because we used the filter-based approach in both stages: before using the GA and during the GA's execution – by using a filter-based fitness function. In addition, we focus on multi-label classification problems, rather than on single-label classification problems as usual in the literature.

Algorithm 6.7: LexGA Tournament Selection in LexGA-ML-CFS()

INPUT: indPool, SEmerit, SEk

SET: sortedPool=null

repeat

- $merit1^{st}ind.$ ← ind. with larger merit
- $merit2^{nd}ind.$ ← ind. with smaller merit
- **if** $merit1^{st} - merit2^{nd} > SEmerit$
  - **then** Select 1st ind. and put it into sorted Pool
  - **else**
    - $k1^{st}ind.$ ← ind. with smaller k
    - $k2^{nd}ind.$ ← ind. with larger k
    - **if** $k2^{nd} - k1^{st} > SEk$ **and** $merit1^{st} - merit2^{nd} > 0.5 * SEmerit$
      - **then** Select 1st ind. and put it into sorted Pool
      - **else** Select ind with larger merit
- Remove selected ind. from indPool

until $ind.Pool = EmptySet$

6.3 ML-CFS with a Lexicographic Multi-Objective Genetic Algorithm (LexGA-ML-CFS)

In this Section, we propose a more sophisticated multi-objective GA based on the lexicographic approach as a new search method for our ML-CFS method. The lexicographic approach assigns different priorities to different objectives (evaluation criteria), and then it focuses on optimizing the objectives in decreasing order of priority. Each evaluation criterion is treated separately, and is used to measure a different aspect of quality of a candidate solution. In essence, the lexicographic multi-objective evaluation works as follows. When comparing two candidate solu-

tions (individuals), first the solutions are compare with respect to the first, highest priority objective. If one solution is significantly better than the other, the former is chosen. Otherwise, the two solutions are compared with respect to the second objective. If one solution is significantly better, then that solution is chosen. Otherwise, there is no significant difference between the two solutions according to both objectives, and in this case the solution with the best value of the first objective is chosen. Hence, this approach avoids the problems of combining different evaluation criteria into a single weighted formula with different weights assigned to different criteria, in particular the problem that the weights are usually specified in an arbitrary, ad-hoc fashion by the user [36] - see also Chapter 2.

In GA-ML-CFS (described in Section 6.2) and LexGA-ML-CFS, the population initialization, crossover and mutation operators are the same. There are only two main different steps between these two types of GA, which are the approach used to evaluate each individual and the approach used to select the winner in the tournament selection step. In LexGA-ML-CFS, the fitness of an individual is evaluated based on two criteria: (1) the merit function, which is shown in Equation 6.1; and (2) the number of selected features (k). For multi-objective tournament selection, we use the standard error of the merit (SEmerit) and the standard error of the number of selected features (SEk), where the standard errors are calculated across all individuals in the individual pool.

The pseudocode of LexGA tournament selection is shown in Algorithm 6.7. When comparing two candidate individuals (feature subsets), if the difference between the merit values of the two individuals is greater than the standard error of the merit (SEmerit) across all individuals in the current population, the individual with the greater merit value is chosen as the tournament's winner. Otherwise, if the difference of the k value of the individual with larger k (more selected features) minus the k value of the individual with smaller k (fewer features) is greater than the standard error of k (SEk) across all individuals in the current population and

the difference of the merit value of the individual with greater merit minus the merit value of the individual with smaller merit is larger than half the SEmerit, then the individual with smallest k (smallest feature subset) is chosen. Otherwise, the individual with the largest merit is chosen.

Our preliminary experiments showed that a lexicographic optimization tournament using only a condition on the difference in k values as the second criterion was leading the GA to select individuals based on this second lexicographic criterion (after a tie being observed in the first criterion) very often, leading the GA to return solutions that had a relatively small number of features but relatively poor predictive accuracy. To prevent the GA from selecting individuals based on the second lexicographic criterion without a second thought about the merit value of each individual, the second condition in the above otherwise, if statement (the condition for the difference in merit to be greater than half the SEmerit) was added. Hence, the addition of this second condition based on merit, when evaluating the second lexicographic criterion, helps to de-emphasize the importance of the second lexicographic objective (minimizing the number of selected features), which therefore helps to emphasize the importance of the first lexicographic objective (maximizing predictive accuracy).

6.4 Datasets Used in the Experiments

For the GA-ML-CFS and LexGA-ML-CFS experiments, we used 14 datasets (shown in Table 6.1), which were obtained from the multi-label dataset repository website `http://mulan.sourceforge.net/datasets.html`[30]. For all datasets mentioned in Table 6.1, we used the predefined partition of each dataset into training and test sets provided by the MULAN repository.

We separated all datasets into two groups: (1) datasets for parameter optimiza-

Table 6.1: Datasets used in the experiments

Dataset Symbol	Dataset Name	Dataset Description					
		Instances	**Features**	**Labels**	**Cardinality**	**Density**	**Distinct**
Parameter Optimization Datasets							
N1	CAL500	502	68	174	26.044	0.150	502
N2	Scene	2407	294	6	1.074	0.179	15
N3	Emotions	593	72	6	1.869	0.311	27
N4	Yeast	2417	103	14	4.237	0.303	198
Evaluation Datasets							
N5	Business	11314	21924	30	1.600	0.053	158
N6	Art	7484	23146	26	1.659	0.063	404
N7	Education	12030	27534	33	1.455	0.044	348
N8	Recreation	12828	30324	22	1.428	0.065	369
N9	Health	9205	30635	32	1.635	0.051	235
N10	Entertainment	12730	32001	21	1.405	0.067	246
N11	Computer	12444	34096	33	1.518	0.046	296
N12	Science	6428	37187	40	1.471	0.037	332
B1	Enron	1702	1001	53	3.378	0.064	753
B2	Medical	978	1449	45	1.245	0.028	94

tion and (2) datasets for evaluating our genetic algorithm-based multi-label correlation based feature selection (GA-ML-CFS and LexGA-ML-CFS) algorithms. The parameter optimization group includes 4 datasets; all datasets considered as relatively small (where the number of features is less than 300), while all evaluation datasets have a number of features greater than 1,000. Note that these two groups of datasets were also used in Chapter 4.

6.5 Experimental Methodology

There are two main steps in our experimental methodology to use GA-ML-CFS: (1) finding the best GA parameter setting specifically for each of the two multi-label classification algorithms (i.e. ML-kNN or BPMLL) used in our experiments; (2) running GA-ML-CFS using the parameter setting obtained from step (1) and passing the selected features to the corresponding kind of multi-label classification algorithm. The details of each step are described below:

Step 1: Finding the best parameter setting can be done in two different ways:

Table 6.2: Range of possible settings for each of 6 parameter of the GA-ML-CFS

Parameters	**Tried Settings**
population size (PopSize)	100, 150, 200, 250
number of generations (MaxGen)	50, 100, 150, 200
elitism size (Elite)	2, 4, 6, 8
tournament size (TourSize)	2, 4, 6, 8
crossover probability (GeneCrossProb)	0.2, 0.3, 0.4, 0.5
mutation probability (GeneMuteProb)	0.0025, 0.005, 0.001, 0.01

Table 6.3: GA-ML-CFS' Parameter Setting for The Parameter Optimization Process

	Parameters					
No.	**Pop Size**	**Max Gen**	**Elite Size**	**Tour Size**	**Gene CrossProb**	**Gene MuteProb**
PS01	200	100	2	2	0.5	0.01
PS02	100	100	2	2	0.5	0.01
PS03	150	100	2	2	0.5	0.01
PS04	250	100	2	2	0.5	0.01
PS05	200	50	2	2	0.5	0.01
PS06	200	150	2	2	0.5	0.01
PS07	200	200	2	2	0.5	0.01
PS08	200	100	4	2	0.5	0.01
PS09	200	100	6	2	0.5	0.01
PS10	200	100	8	2	0.5	0.01
PS11	200	100	2	4	0.5	0.01
PS12	200	100	2	6	0.5	0.01
PS13	200	100	2	8	0.5	0.01
PS14	200	100	2	2	0.4	0.01
PS15	200	100	2	2	0.3	0.01
PS16	200	100	2	2	0.2	0.01
PS17	200	100	2	2	0.5	0.005
PS18	200	100	2	2	0.5	0.0025
PS19	200	100	2	2	0.5	0.001

(1) the wrapper-like approach; and (2) the filter approach. In the wrapper-like approach, for each candidate GA parameter setting, the GA with that setting is run in a way analogous to the wrapper approach using the accuracy of the classification algorithm as the quality of the corresponding GA parameter setting. Then, the solution (parameter setting) with the highest predictive accuracy on average, over the four parameter optimization datasets, will be selected. We call this approach "wrapper-like", rather than "wrapper", because it evaluates the quality of individuals by running the target classifier on datasets completely different from the datasets where the GA will be evaluated. This is in contrast to a traditional wrapper approach, which evaluates the quality of individuals by running the target classifier on the same dataset where the GA will be evaluated.

Note that the parameter optimization datasets are in general much smaller (particularly in terms of the number of features) than the evaluation datasets – see Table 6.1. Hence, the wrapper-like approach avoids the wrapper approach's problem of being too computationally expensive for very large datasets. In addition, this "wrapper-like" approach produces recommended parameter settings that are relatively robust (since they were obtained by averaging results from 4 different datasets), so that they can be used as a kind of "default" parameter settings in the experiment with the 10 evaluation datasets, avoiding the time-consuming approach of optimizing parameters for each evaluation dataset separately.

In the filter approach for parameter optimization, for each candidate GA parameter setting, the merit value of that setting was calculated by Equation 6.1. Then the best parameter setting, i.e, the one with highest merit will be selected. In this approach, we find a parameter setting optimized for GA-ML-CFS regardless of the type of classifier to be used later. The effectiveness of these two approaches will be compared later, in Step 2.

We considered 6 GA parameters, each with the range of possible values shown

in Table 6.2. In total, 19 parameter setting combinations were considered (shown in Table 6.3). The default GA parameter setting is PS01, where the PopSize is 200. Then we try three different PopSize values, which are 100, 150 and 250. We did the same approach for all GA parameters, adding about three different values for each parameter in turn. We decided not to use all possible combinations of all GA parameter values because there would be too many parameter settings to be tried. In the parameter optimization step, the size of GA individuals is given by the number of features in the dataset used in the experiment; for example, the individual size is equal to 68 and 294 on CAL500 and Scene datasets respectively.

Step 2: running GA-ML-CFS on the evaluation datasets using the parameter settings obtained in the parameter optimization experiments. In this step, we run four types of experiments: (1) running GA-ML-CFS using parameters optimized by the wrapper-like approach for the use of ML-kNN; (2) running GA-ML-CFS using parameters optimized by the wrapper-like approach for the use of BPMLL, (3) running GA-ML-CFS using parameters optimized by the merit-based filter approach (independent from the classifer); and (4) running other baseline multi-label feature selection methods.

Since all evaluation datasets have a large number of features (varying from 1,001 to 37,187 – see Table 6.1), in this step we use a univariate filter approach to select a subset of the most relevant features before running GA, as discussed in Chapter 4. We did experiments where the number of features selected by the univariate filter method (and therefore the GA's individuals' length) varied between 100, 200, 300 and 400.

Recall that, as discussed in Chapter 4, the motivation for applying this univariate filter method in a pre-processing phase is to reduce the number of candidate features or individual length for the GA when the number of features is very large (as it is the case for the evaluation datasets), in order to reduce the processing

time and improve the scalability of GA-ML-CFS. This approach is often used in the literature on GA for feature selection [15, 108, 118]

Note that in Chapter 4 we have reported results clearly showing that gmiML-CFS obtained in general much better predictive accuracy than the other versions of ML-CFS, both when using ML-kNN and when using BPMLL as the multi-label classifier. Hence, the same approach used to evaluate the correlation between feature subset and class labels of gmiML-CFS was used in GA-ML-CFS. More precisely, the computational results reported next were produced by using Equation 4.11 to calculate the correlation between a feature and all class labels and then evaluating the quality of a candidate feature subset using Equations 6.2 and 6.1, respectively, to calculate $\overline{r_{FL}}$ and the merit of a candidate feature subset F.

6.6 Results for Parameter Optimization of GA-ML-CFS and LexGA-ML-CFS

We first report results obtained by the wrapper-like approach for parameter optimization. After running GA-ML-CFS using 19 parameter settings on the 4 parameter optimization datasets, the feature subset selected by GA for each parameter setting was evaluated by measuring the predictive accuracy of ML-kNN and BPMLL when using that feature subset. As discussed earlier, due to the complexity of multi-label classification, no single predictive accuracy measure is enough to capture different aspects of multi-label classification [18, 112]. Hence, five different popular measures of multi-label predictive accuracy were used in our experiment: Average Precision (Avg.Pre), which is to be maximized, while Coverage (Cov.), Hamming Loss (H.Loss), One-error (One-Err) and Ranking Loss (R.Loss) are to be minimized. All those measures are discussed in [112]. Then we compute the rank of each GA parameter setting for each dataset and each predictive accuracy

Table 6.4: Summary of Ranking Results for Parameter Setting Optimization with the wrapper-like approach using the ML-KNN classifier

Para.Set	**Overall ranking (over 5 accuracy measures) per dataset**				**Mean rank over the 4 datasets**
	Emotion	**CAL501**	**Yeast**	**Scene**	
PS1	13.13	6.00	12.50	9.00	10.16
PS2	14.88	10.50	4.00	10.25	9.91
PS3	13.00	13.38	11.25	11.00	12.16
PS4	9.75	9.00	4.00	15.50	9.56
PS5	10.50	15.00	3.00	10.00	9.63
PS6	11.25	6.00	12.75	7.50	9.38
PS7	10.25	9.00	16.50	14.75	12.63
PS8	10.00	12.25	10.00	6.50	9.69
PS9	6.00	13.75	9.00	13.25	10.50
PS10	13.50	9.63	9.38	2.25	8.69
PS11	12.00	6.75	15.75	7.00	10.38
PS12	9.75	11.00	5.25	6.25	8.06
PS13	11.75	14.63	12.13	7.25	11.44
PS14	2.50	9.13	3.75	3.00	**4.59**
PS15	6.88	8.00	9.00	16.75	10.16
PS16	9.00	7.75	13.00	17.50	11.81
PS17	11.88	8.25	12.25	9.00	10.34
PS18	8.25	11.75	12.25	17.00	12.31
PS19	5.75	8.25	14.25	6.25	8.63

measure. That is, the GA parameter setting with the best value of a given accuracy measure is assigned rank 1, and the worst parameter setting is assigned rank 19, for each combination of dataset and accuracy measure. Next, for each dataset, we produced a ranking of the 19 parameter settings by computing the average of their rank across the five accuracy measures. Finally, we produced the overall ranking of the 19 parameter settings by averaging the previously computed rank across all 4 datasets used for parameter optimization.

Tables 6.4 and 6.5 show the overall ranking of each GA-ML-CFS' parameter setting over all evaluation measures for each dataset and the overall rank across the 4 datasets used for parameter optimization for the MLkNN and BPMLL classification algorithm, respectively, using the wrapper-like approach for parameter optimization. The best parameter setting for GA-ML-CFS, which will be used in the experiments to evaluate the predictive accuracy of GA-ML-CFS for ML-kNN, is PS14 (with the best rank of 4.59 in Table 6.4); while the best parameter setting

Table 6.5: Summary of Ranking Results for Parameter Setting Optimization with the wrapper-like approach using the BPMLL classifier

Para.Set	**Overall ranking (over 5 accuracy measures) per dataset**				**Mean rank over the 4 datasets**
	Emotion	**CAL501**	**Yeast**	**Scene**	
PS1	6.00	10.75	5.25	12.25	8.56
PS2	7.00	9.13	6.50	15.75	9.59
PS3	2.50	3.38	10.50	3.00	**4.84**
PS4	7.88	10.25	10.75	13.75	10.66
PS5	18.50	10.38	11.50	5.25	11.41
PS6	8.50	10.00	9.25	9.00	9.19
PS7	1.25	15.75	15.75	8.75	10.38
PS8	8.88	13.88	8.50	11.25	10.63
PS9	4.00	3.63	15.00	15.50	9.53
PS10	12.00	8.00	7.25	2.25	7.38
PS11	9.25	7.75	5.00	9.00	7.75
PS12	10.50	8.00	12.50	11.00	10.50
PS13	16.75	8.00	10.25	12.50	11.88
PS14	12.38	7.00	13.25	3.50	9.03
PS15	12.25	11.63	2.25	5.25	7.84
PS16	17.00	15.75	10.25	2.75	11.44
PS17	13.13	13.50	7.25	12.25	11.53
PS18	12.00	9.88	13.50	18.00	13.34
PS19	10.25	13.38	15.50	19.00	14.53

Table 6.6: Summary of Ranking Results for Merit-Based Parameter Setting Optimization with the filter approach for GA-ML-CFS

Para. Set	**Overall ranking (Merit values) per dataset**				**Mean rank over 4 datasets**
	CAL500	**Emotion**	**Scene**	**Yeast**	
PS01	11	9	5	10	8.75
PS02	18	16	18	18	17.5
PS03	3	17	15	13	12
PS04	3	10	4	12	7.25
PS05	19	19	19	19	19
PS06	3	8	3	8	5.5
PS07	1	5	1	7	**3.5**
PS08	8	14	7	14	10.75
PS09	9	13	11	11	11
PS10	14	15	2	15	11.5
PS11	16	2	12	5	8.75
PS12	13	2	13	2	7.5
PS13	5	2	14	1	5.5
PS14	7	11	9	9	9
PS15	15	12	10	16	13.25
PS16	10	18	6	17	12.75
PS17	6	7	8	6	6.75
PS18	17	4	16	4	10.25
PS19	12	6	17	3	9.5

Table 6.7: Summary of Ranking Results for Parameter Setting Optimization for LexGA-MLCFS with the wrapper-like approach using the ML-KNN classifier

Para.Set	**Overall ranking (over 5 accuracy measures) per dataset**				**Mean rank over the 4 datasets**
	Emotion	**CAL501**	**Yeast**	**Scene**	
PS01	9.00	8.13	9.00	11.00	9.28
PS02	9.00	4.38	10.75	4.75	**7.22**
PS03	7.63	10.38	14.00	10.00	10.50
PS04	9.00	10.63	9.00	2.25	7.72
PS05	10.00	14.50	13.25	5.00	10.69
PS06	9.00	9.63	13.75	3.00	8.84
PS07	9.00	10.38	10.75	7.38	9.38
PS08	9.00	10.63	12.00	7.50	9.78
PS09	9.00	12.13	3.75	14.25	9.78
PS10	9.00	10.38	7.00	6.63	8.25
PS11	9.00	9.50	9.25	14.00	10.44
PS12	9.00	6.38	11.25	18.75	11.34
PS13	9.00	11.88	7.75	16.75	11.34
PS14	13.75	10.63	10.50	12.00	11.72
PS15	14.88	10.63	10.50	1.25	9.31
PS16	9.00	9.63	7.75	8.25	8.66
PS17	7.38	9.50	8.25	12.75	9.47
PS18	13.25	10.38	10.00	16.25	12.47
PS19	15.13	10.38	11.50	18.25	13.81

for BPMLL is PS03 (with the best rank of 4.48 in Table 6.5).

Table 6.6 shows the overall ranking of each GA-ML-CFS' parameter setting over all evaluation measures for each dataset and the overall rank across the 4 datasets used for parameter optimization, when using the filter approach for parameter optimization, where parameter settings are optimized in a way independent from the classification algorithm. According to this approach, the parameter setting for GA-ML-CFS is PS07.

The best parameter setting for LexGA-ML-CFS, which will be used in the experiments to evaluate the predictive accuracy of LexGA-ML-CFS for ML-kNN, is PS02 (see Table 6.7); while the best parameter setting for BPMLL is PS15 (see Table 6.8). Note that those two parameter settings were chosen based on the wrapper-like approach for parameter optimization. Moreover the best parameter setting for LexGA-ML-CFS based on the filter approach is PS10 (see Table 6.9).

Table 6.8: Summary of Ranking Results for Parameter Setting Optimization for LexGA-MLCFS with the wrapper-like approach using the BPMLL classifier

Para.Set	**Overall ranking (over 5 accuracy measures) per dataset**				**Mean rank over the 4 datasets**
	Emotion	**CAL501**	**Yeast**	**Scene**	
PS01	8.00	7.13	6.25	11.25	8.16
PS02	10.00	9.63	11.50	4.00	8.78
PS03	17.00	10.00	8.50	2.75	9.56
PS04	4.50	16.00	16.25	3.50	10.06
PS05	18.25	14.88	3.25	10.25	11.66
PS06	9.00	11.38	10.50	4.00	8.72
PS07	8.50	9.38	11.25	4.75	8.47
PS08	8.75	15.25	12.75	7.75	11.13
PS09	16.00	6.13	13.00	8.00	10.78
PS10	7.50	3.50	8.25	7.75	6.75
PS11	10.00	10.00	13.75	15.50	12.31
PS12	8.25	10.88	12.25	15.50	11.72
PS13	13.25	12.75	12.75	17.75	14.13
PS14	12.00	4.00	7.25	7.25	7.63
PS15	4.00	9.75	6.00	6.75	**6.63**
PS16	14.75	9.13	7.50	13.00	11.09
PS17	6.25	5.25	8.50	14.00	8.50
PS18	10.50	9.25	11.25	17.25	12.06
PS19	3.50	15.75	9.25	19.00	11.88

Table 6.9: Summary of Ranking Results for Merit-Based Parameter Setting Optimization with the filter approach for LexGA-ML-CFS

Para. Set	**Overall ranking (Merit values) per dataset**				**Mean rank over 4 datasets**
	Emotions	**Yeast**	**CAL500**	**Scene**	
PS01	10	6	6	2	5.88
PS02	10	13	3	12	9.38
PS03	10	2	15	4	7.63
PS04	10	11	19	13	13.13
PS05	10	1	11	1	5.63
PS06	10	10	16	5	10.13
PS07	10	5	4	8	6.50
PS08	10	3	13	11	9.13
PS09	10	5	7	7	7.00
PS10	10	8	1	3	**5.38**
PS11	10	17	9	6	10.38
PS12	10	9	2	18	9.63
PS13	10	16	14	9	12.13
PS14	10	7	18	15	12.38
PS15	10	12	10	17	12.13
PS16	10	19	17	10	13.88
PS17	10	15	5	16	11.38
PS18	19	14	12	14	14.75
PS19	10	18	8	19	13.63

The details of each parameter setting mentioned above are shown in Table 6.3.

6.7 Results for GA-ML-CFS and LexGA-ML-CFS on Evaluation Datasets

Subsections 6.7.1 and 6.7.2 report results comparing four versions of GA-ML-CFS, namely: (1) a single-objective GA-ML-CFS using parameter setting optimized by the filter approach (gmiGA-filt), (2) a multi-objective GA-ML-CFS using parameter setting optimized by the filter approach (gmiLexGA-filt), (3) a single-objective GA-ML-CFS using parameter setting optimized by the wrapper-like approach (gmiGA-wrap); and (4) a multi-objective GA-ML-CFS using parameter setting optimized by the wrapper-like approach (gmiLexGA-wrap). The classifier used was ML-kNN in Subsection 6.7.1 and BPMLL in Subsection 6.7.2. In both these Subsections the GA-ML-CFS version used in the experiments was the one using mutual information for class label weighting and absolute value of the correlation coefficient, since this version clearly obtained better results than other versions of ML-CFS in Chapter 4. More precisely, GA-ML-CFS versions used in this current Chapter evaluate the quality of feature subset in the same way as the ML-CFS version where class labels with greater MI (Mutual Information) are assigned greater weights (gmi).

6.7.1 ML-kNN's Results for GA-ML-CFS and LexGA-ML-CFS Using Mutual Information for Class Label Weighting with two parameter optimization approaches: wrapper-like approach versus filter approach

All GA results are an average over 5 runs with a different random seed used to create the initial population in each run. In Tables 6.10 - 6.13, each column "R" shows the rank ("1" is better than "2") of each method (GA-wrap. and GA-filt.) for each dataset according to the accuracy measure on the corresponding left column. The last column reports the average rank of each method across all five accuracy measures, for each dataset. The last row reports the average rank for each column (across all 10 datasets).

Recall that, in these Tables, gmiGA-wrap denotes gmiGA-ML-CFS with parameter setting optimized by the wrapper-like approach (PS14 in Table 6.3) and gmiGA-filt denotes gmiGA-ML-CFS with parameter optimized by the filter approach (PS07 in Table 6.3). For the LexGA version, gmiLexGA-wrap denotes gmiLexGA-ML-CFS with parameter setting optimized by the wrapper-like approach (PS02 in Table 6.3) and gmiLexGA-filt denotes gmiLexGA-ML-CFS with parameter optimized by the filter approach (PS10 in Table 6.3).

To summarize the results, we will focus on the average ranks obtained in the 4 GA-ML-CFS versions across all datasets and all 5 accuracy measures. When using ML-kNN as the classifier, gmiGA-wrap obtained better predictive accuracy (lower average rank) than gmiLexGA-wrap, gmiGA-filt, and gmiLexGA-filt in the experiments with individual length of 100, with average rank of 2.33 versus 2.40, 2.57 and 2.70, respectively (Table 6.10).

When the individual length is 200 (Table 6.11), gmiGA-wrap again outper-

formed other versions of GA-ML-CFS with the smallest average rank (2.08); while gmiGA-filt, gmiLexGA-wrap, and gmiLexGA-filt obtained the larger average ranks 2.32, 2.69 and 2.91, respectively.

When the individual length is 300 (Table 6.12), gmiGA-filt obtained better predictive accuracy (lower average rank) than gmiGA-wrap, gmiLexGA-wrap; and gmiLexGA-filt in the experiments with individual length of 300, with average rank of 1.86 versus 1.90, 2.96 and 3.28, respectively (Table 6.12).

Moreover, when the individual length is 400 (Table 6.13), gmiGA-filt again outperformed other versions of GA-ML-CFS with the smallest average rank (2.00) while gmiGA-wrap, gmiLexGA-wrap; and gmiLexGA-filt obtained the larger average ranks 2.12, 2.92 and 2.92, respectively.

Table 6.14 shows the number and percentage of selected features and average rank over the 10 datasets for each GA individual length. Clearly, when the individual length equals to 100 and 200, gmiGA-wrap obtained the best overall average rank in term of accuracy (2.33 and 2.08), and selected 26.90 % and 23.37 % of the features in the GA's feature space. When the individual length is larger (300 and 400) gmiGA-filt obtained the best predictive accuracy in these two cases (average rank of 1.86 and 2.00, respectively) with 18.15 % and 18.35 % of the selected features. Overall (last row of Table 6.14), gmiGA-wrap obtained the best average rank (2.11) and the second smallest percentage of selected features (24.81%); and so it was the best version of GA-ML-CFS in these experiments (with ML-kNN), since maximizing accuracy is more important than minimizing the number of selected features.

Figure 6.1 shows the overall average ranking (AR) for the four versions of GA-ML-CFS investigated in this section plotted against average size of selected features across four feature space sizes using ML-kNN as the classifier. Clearly,

Table 6.10: Predictive accuracy for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) with ML-kNN Classifier (individual length = 100)

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiGA-filt	0.584	3.0	13.763	1.0	0.058	2.0	0.395	4.0	0.101	2.0	2.40
	gmiLexGA-filt	0.584	1.0	13.784	3.0	0.058	3.0	0.392	2.0	0.100	3.0	2.40
	gmiGA-wrap	0.584	2.0	13.763	2.0	0.057	1.0	0.394	3.0	0.100	4.0	2.40
	gmiLexGA-wrap	0.581	4.0	13.858	4.0	0.058	4.0	0.390	1.0	0.101	1.0	2.80
Medical	gmiGA-filt	0.794	1.0	3.065	1.0	0.015	2.0	0.256	1.0	0.049	4.0	1.80
	gmiLexGA-filt	0.780	4.0	3.125	4.0	0.016	4.0	0.275	4.0	0.050	1.0	3.40
	gmiGA-wrap	0.793	2.0	3.111	2.0	0.015	1.0	0.256	2.0	0.050	2.0	1.80
	gmiLexGA-wrap	0.788	3.0	3.119	3.0	0.015	3.0	0.267	3.0	0.050	3.0	3.00
Business	gmiGA-filt	0.874	4.0	2.386	2.0	0.028	4.0	0.124	4.0	0.043	2.5	3.30
	gmiLexGA-filt	0.874	2.0	2.392	4.0	0.028	2.0	0.123	2.0	0.043	1.0	2.20
	gmiGA-wrap	0.874	1.0	2.386	1.0	0.028	2.0	0.124	3.0	0.043	2.5	1.90
	gmiLexGA-wrap	0.874	3.0	2.389	3.0	0.028	2.0	0.123	1.0	0.043	4.0	2.60
Art	gmiGA-filt	0.527	3.0	5.409	2.0	0.059	4.0	0.588	1.0	0.150	4.0	2.80
	gmiLexGA-filt	0.525	4.0	5.426	4.0	0.059	3.0	0.591	4.0	0.151	1.0	3.20
	gmiGA-wrap	0.527	2.0	5.409	1.0	0.059	1.0	0.589	2.0	0.150	3.0	1.80
	gmiLexGA-wrap	0.527	1.0	5.410	3.0	0.059	2.0	0.590	3.0	0.150	2.0	2.20
Education	gmiGA-filt	0.543	4.0	3.919	1.0	0.042	2.0	0.605	4.0	0.092	2.0	2.60
	gmiLexGA-filt	0.543	3.0	3.943	4.0	0.042	4.0	0.604	3.0	0.093	1.0	3.00
	gmiGA-wrap	0.544	2.0	3.924	3.0	0.042	3.0	0.600	2.0	0.092	3.0	2.60
	gmiLexGA-wrap	0.545	1.0	3.922	2.0	0.041	1.0	0.599	1.0	0.092	4.0	1.80
Recreation	gmiGA-filt	0.536	1.0	4.307	2.0	0.058	1.0	0.601	1.0	0.158	3.0	1.60
	gmiLexGA-filt	0.535	3.0	4.330	4.0	0.059	4.0	0.603	3.0	0.158	1.0	3.00
	gmiGA-wrap	0.536	2.0	4.286	1.0	0.059	2.0	0.603	2.0	0.157	4.0	2.20
	gmiLexGA-wrap	0.534	4.0	4.318	3.0	0.059	3.0	0.605	4.0	0.158	2.0	3.20
Health	gmiGA-filt	0.631	4.0	3.791	4.0	0.049	4.0	0.479	4.0	0.075	1.0	3.40
	gmiLexGA-filt	0.631	2.0	3.783	1.0	0.049	2.0	0.477	2.0	0.075	2.0	1.80
	gmiGA-wrap	0.631	3.0	3.787	3.0	0.049	3.0	0.478	3.0	0.075	4.0	3.20
	gmiLexGA-wrap	0.632	1.0	3.784	2.0	0.049	1.0	0.476	1.0	0.075	3.0	1.60
Ent.ment	gmiGA-filt	0.597	3.0	3.152	3.0	0.055	1.0	0.543	3.0	0.118	2.0	2.40
	gmiLexGA-filt	0.595	4.0	3.159	4.0	0.056	4.0	0.544	4.0	0.119	1.0	3.40
	gmiGA-wrap	0.600	1.0	3.151	2.0	0.055	2.0	0.539	1.0	0.118	4.0	2.00
	gmiLexGA-wrap	0.598	2.0	3.146	1.0	0.055	3.0	0.542	2.0	0.118	3.0	2.20
Computer	gmiGA-filt	0.625	4.0	4.390	1.0	0.040	2.0	0.444	4.0	0.093	3.0	2.80
	gmiLexGA-filt	0.625	3.0	4.413	4.0	0.040	3.0	0.442	3.0	0.094	1.0	2.80
	gmiGA-wrap	0.625	2.0	4.393	3.0	0.040	1.0	0.442	2.0	0.093	2.0	2.00
	gmiLexGA-wrap	0.626	1.0	4.390	2.0	0.040	4.0	0.441	1.0	0.093	4.0	2.40
Science	gmiGA-filt	0.459	2.0	6.931	1.0	0.035	3.0	0.670	3.0	0.136	4.0	2.60
	gmiLexGA-filt	0.460	1.0	6.988	4.0	0.035	2.0	0.666	1.0	0.137	1.0	1.80
	gmiGA-wrap	0.458	4.0	6.937	2.0	0.035	4.0	0.672	4.0	0.136	3.0	3.40
	gmiLexGA-wrap	0.459	3.0	6.972	3.0	0.035	1.0	0.668	2.0	0.137	2.0	2.20
MEAN	gmiGA-filt		2.9		1.8		2.5		2.9		2.8	2.57
	gmiLexGA-filt		2.7		3.6		3.1		2.8		1.3	2.70
	gmiGA-wrap		2.1		2.0		2.0		2.4		3.2	2.33
	gmiLexGA-wrap		2.3		2.6		2.4		1.9		2.8	2.40

Table 6.11: Predictive accuracy for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) with ML-kNN Classifier (individual length = 200)

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiGA-filt	0.581	4.0	13.705	4.0	0.058	2.0	0.396	3.0	0.100	4.0	3.40
	gmiLexGA-filt	0.585	3.0	13.511	2.0	0.058	3.0	0.392	2.0	0.099	3.0	2.60
	gmiGA-wrap	0.585	2.0	13.570	3.0	0.058	4.0	0.397	4.0	0.098	2.0	3.00
	gmiLexGA-wrap	0.587	1.0	13.469	1.0	0.058	1.0	0.390	1.0	0.098	1.0	1.00
Medical	gmiGA-filt	0.810	1.0	2.893	1.0	0.016	1.0	0.236	1.0	0.045	1.0	1.00
	gmiLexGA-filt	0.808	2.0	2.896	2.0	0.016	3.0	0.239	2.0	0.046	2.0	2.20
	gmiGA-wrap	0.797	3.0	3.017	3.0	0.016	4.0	0.254	3.0	0.048	3.0	3.20
	gmiLexGA-wrap	0.795	4.0	3.042	4.0	0.016	2.0	0.254	4.0	0.049	4.0	3.60
Business	gmiGA-filt	0.873	2.0	2.333	4.0	0.028	2.0	0.126	4.0	0.042	3.0	3.00
	gmiLexGA-filt	0.873	4.0	2.330	2.0	0.028	1.0	0.125	1.0	0.041	2.0	2.00
	gmiGA-wrap	0.873	3.0	2.331	3.0	0.028	3.0	0.126	2.0	0.042	4.0	3.00
	gmiLexGA-wrap	0.874	1.0	2.322	1.0	0.028	4.0	0.126	3.0	0.041	1.0	2.00
Art	gmiGA-filt	0.533	3.0	5.336	2.0	0.059	4.0	0.584	2.0	0.148	4.0	3.00
	gmiLexGA-filt	0.531	4.0	5.343	4.0	0.059	3.0	0.586	4.0	0.148	3.0	3.60
	gmiGA-wrap	0.536	1.0	5.324	1.0	0.059	1.0	0.578	1.0	0.147	1.0	1.00
	gmiLexGA-wrap	0.533	2.0	5.337	3.0	0.059	2.0	0.584	3.0	0.147	2.0	2.40
Education	gmiGA-filt	0.555	1.0	3.861	1.0	0.041	1.0	0.584	1.0	0.091	1.0	1.00
	gmiLexGA-filt	0.544	4.0	3.923	4.0	0.041	4.0	0.599	4.0	0.093	4.0	4.00
	gmiGA-wrap	0.551	2.0	3.896	2.0	0.041	2.0	0.588	2.0	0.092	2.0	2.00
	gmiLexGA-wrap	0.547	3.0	3.910	3.0	0.041	3.0	0.595	3.0	0.092	3.0	3.00
Recreation	gmiGA-filt	0.572	2.0	4.198	4.0	0.055	1.0	0.544	1.0	0.152	3.0	2.20
	gmiLexGA-filt	0.571	4.0	4.191	3.0	0.055	2.0	0.549	4.0	0.152	4.0	3.40
	gmiGA-wrap	0.572	1.0	4.171	1.0	0.055	3.0	0.545	2.0	0.151	1.0	1.60
	gmiLexGA-wrap	0.571	3.0	4.174	2.0	0.055	4.0	0.546	3.0	0.151	2.0	2.80
Health	gmiGA-filt	0.686	2.0	3.426	4.0	0.042	3.0	0.388	2.0	0.064	4.0	3.00
	gmiLexGA-filt	0.685	3.0	3.416	3.0	0.043	4.0	0.393	4.0	0.064	2.0	3.20
	gmiGA-wrap	0.687	1.0	3.411	1.0	0.042	1.0	0.387	1.0	0.064	1.0	1.00
	gmiLexGA-wrap	0.685	4.0	3.411	2.0	0.042	2.0	0.393	3.0	0.064	3.0	2.80
Ent.ment	gmiGA-filt	0.615	2.0	3.088	3.0	0.053	1.0	0.508	3.0	0.112	3.0	2.40
	gmiLexGA-filt	0.613	3.0	3.074	2.0	0.054	4.0	0.504	2.0	0.112	2.0	2.60
	gmiGA-wrap	0.618	1.0	3.056	1.0	0.054	2.5	0.498	1.0	0.111	1.0	1.30
	gmiLexGA-wrap	0.610	4.0	3.095	4.0	0.054	2.5	0.510	4.0	0.113	4.0	3.70
Computer	gmiGA-filt	0.641	3.0	4.221	4.0	0.038	1.0	0.429	1.0	0.090	4.0	2.60
	gmiLexGA-filt	0.641	2.0	4.206	3.0	0.039	4.0	0.431	4.0	0.090	3.0	3.20
	gmiGA-wrap	0.643	1.0	4.178	1.0	0.038	3.0	0.430	2.0	0.089	1.0	1.60
	gmiLexGA-wrap	0.641	4.0	4.200	2.0	0.038	2.0	0.430	3.0	0.089	2.0	2.60
Science	gmiGA-filt	0.485	1.0	6.749	2.0	0.034	2.0	0.636	1.0	0.132	2.0	1.60
	gmiLexGA-filt	0.481	3.0	6.705	1.0	0.034	3.5	0.646	3.0	0.131	1.0	2.30
	gmiGA-wrap	0.482	2.0	6.792	4.0	0.034	3.5	0.638	2.0	0.133	4.0	3.10
	gmiLexGA-wrap	0.471	4.0	6.781	3.0	0.034	1.0	0.657	4.0	0.132	3.0	3.00
MEAN	gmiGA-filt		2.1		2.9		1.8		1.9		2.9	2.32
	gmiLexGA-filt		3.2		2.6		3.2		3.0		2.6	2.91
	gmiGA-wrap		1.7		2.0		2.7		2.0		2.0	2.08
	gmiLexGA-wrap		3.0		2.5		2.4		3.1		2.5	2.69

Table 6.12: Predictive accuracy for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) with ML-kNN Classifier (individual length = 300)

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiGA-filt	0.585	3.0	13.414	3.0	0.05776	3.0	0.396	4.0	0.097	2.0	3.00
	gmiLexGA-filt	0.586	2.0	13.415	4.0	0.058	4.0	0.394	3.0	0.097	3.0	3.20
	gmiGA-wrap	0.590	1.0	13.391	2.0	0.05702	1.0	0.382	1.0	0.098	4.0	1.80
	gmiLexGA-wrap	0.585	4.0	13.345	1.0	0.057	2.0	0.391	2.0	0.097	1.0	2.00
Medical	gmiGA-filt	0.809	1.0	2.857	1.0	0.01594	1.0	0.247	3.0	0.045	1.0	1.40
	gmiLexGA-filt	0.783	4.0	3.118	4.0	0.017	4.0	0.279	4.0	0.051	4.0	4.00
	gmiGA-wrap	0.805	2.0	2.899	2.0	0.01622	2.0	0.245	1.0	0.046	2.0	1.80
	gmiLexGA-wrap	0.797	3.0	2.991	3.0	0.017	3.0	0.246	2.0	0.048	3.0	2.80
Business	gmiGA-filt	0.875	1.0	2.285	1.0	0.02818	3.0	0.127	3.0	0.040	1.0	1.80
	gmiLexGA-filt	0.875	3.0	2.309	3.0	0.028	2.0	0.126	2.0	0.041	3.0	2.60
	gmiGA-wrap	0.875	2.0	2.294	2.0	0.0283	4.0	0.126	1.0	0.041	2.0	2.20
	gmiLexGA-wrap	0.874	4.0	2.330	4.0	0.028	1.0	0.127	4.0	0.041	4.0	3.40
Art	gmiGA-filt	0.540	1.0	5.311	3.0	0.0584	1.0	0.576	2.0	0.146	3.0	2.00
	gmiLexGA-filt	0.534	3.0	5.285	2.0	0.058	2.0	0.586	4.0	0.145	2.0	2.60
	gmiGA-wrap	0.533	4.0	5.325	4.0	0.0585	3.0	0.584	3.0	0.146	4.0	3.60
	gmiLexGA-wrap	0.540	2.0	5.220	1.0	0.059	4.0	0.574	1.0	0.143	1.0	1.80
Education	gmiGA-filt	0.560	1.0	3.819	2.0	0.04048	1.0	0.577	1.0	0.089	1.0	1.20
	gmiLexGA-filt	0.551	3.0	3.875	3.0	0.041	3.0	0.591	3.0	0.091	3.0	3.00
	gmiGA-wrap	0.558	2.0	3.817	1.0	0.04056	2.0	0.582	2.0	0.089	2.0	1.80
	gmiLexGA-wrap	0.546	4.0	3.908	4.0	0.041	4.0	0.597	4.0	0.092	4.0	4.00
Recreation	gmiGA-filt	0.586	2.0	4.092	2.0	0.0543	1.0	0.526	1.0	0.147	2.0	1.60
	gmiLexGA-filt	0.580	4.0	4.144	4.0	0.055	3.0	0.535	4.0	0.150	4.0	3.80
	gmiGA-wrap	0.586	1.0	4.072	1.0	0.0546	2.0	0.527	2.0	0.147	1.0	1.40
	gmiLexGA-wrap	0.581	3.0	4.119	3.0	0.055	4.0	0.533	3.0	0.149	3.0	3.20
Health	gmiGA-filt	0.690	1.0	3.378	4.0	0.04252	1.0	0.390	1.0	0.063	2.0	1.80
	gmiLexGA-filt	0.683	4.0	3.378	3.0	0.043	4.0	0.405	4.0	0.063	4.0	3.80
	gmiGA-wrap	0.687	2.0	3.360	1.0	0.04256	2.0	0.397	2.0	0.063	1.0	1.60
	gmiLexGA-wrap	0.684	3.0	3.377	2.0	0.043	3.0	0.400	3.0	0.063	3.0	2.80
Ent.ment	gmiGA-filt	0.625	2.0	3.033	4.0	0.0529	1.0	0.488	1.0	0.110	4.0	2.40
	gmiLexGA-filt	0.624	3.0	2.994	2.0	0.054	4.0	0.498	3.0	0.109	2.0	2.80
	gmiGA-wrap	0.628	1.0	2.971	1.0	0.05378	2.0	0.490	2.0	0.108	1.0	1.40
	gmiLexGA-wrap	0.619	4.0	3.014	3.0	0.054	3.0	0.500	4.0	0.109	3.0	3.40
Computer	gmiGA-filt	0.648	1.0	4.129	1.0	0.0376	2.0	0.426	2.0	0.087	1.0	1.40
	gmiLexGA-filt	0.646	4.0	4.194	4.0	0.038	4.0	0.428	3.0	0.089	4.0	3.80
	gmiGA-wrap	0.647	2.0	4.164	3.0	0.0375	1.0	0.426	1.0	0.088	3.0	2.00
	gmiLexGA-wrap	0.647	3.0	4.154	2.0	0.038	3.0	0.428	4.0	0.088	2.0	2.80
Science	gmiGA-filt	0.480	2.0	6.704	3.0	0.03378	1.0	0.647	2.0	0.131	2.0	2.00
	gmiLexGA-filt	0.475	3.0	6.744	4.0	0.034	2.0	0.648	3.0	0.132	4.0	3.20
	gmiGA-wrap	0.481	1.0	6.628	1.0	0.03388	3.0	0.645	1.0	0.129	1.0	1.40
	gmiLexGA-wrap	0.473	4.0	6.701	2.0	0.034	4.0	0.653	4.0	0.131	3.0	3.40
MEAN	gmiGA-filt		1.5		2.4		1.5		2.0		1.9	1.86
	gmiLexGA-filt		3.3		3.3		3.2		3.3		3.3	3.28
	gmiGA-wrap		1.8		1.8		2.2		1.6		2.1	1.90
	gmiLexGA-wrap		3.4		2.5		3.1		3.1		2.7	2.96

Table 6.13: Predictive accuracy for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) with ML-kNN Classifier (individual length = 400)

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiGA-filt	0.581	3.0	13.551	4.0	0.05794	4.0	0.402	3.0	0.099	4.0	3.60
	gmiLexGA-filt	0.582	2.0	13.450	2.0	0.057	2.0	0.390	2.0	0.098	2.0	2.00
	gmiGA-wrap	0.580	4.0	13.358	1.0	0.05762	3.0	0.406	4.0	0.098	1.0	2.60
	gmiLexGA-wrap	0.584	1.0	13.532	3.0	0.056	1.0	0.382	1.0	0.098	3.0	1.80
Medical	gmiGA-filt	0.801	1.0	2.920	1.0	0.016	1.0	0.252	1.0	0.047	1.0	1.00
	gmiLexGA-filt	0.774	3.0	3.269	3.0	0.018	4.0	0.284	3.0	0.054	3.0	3.20
	gmiGA-wrap	0.796	2.0	3.030	2.0	0.01694	2.0	0.258	2.0	0.049	2.0	2.00
	gmiLexGA-wrap	0.768	4.0	3.337	4.0	0.017	3.0	0.291	4.0	0.055	4.0	3.80
Business	gmiGA-filt	0.876	4.0	2.308	4.0	0.02802	2.0	0.124	2.0	0.041	4.0	3.20
	gmiLexGA-filt	0.877	2.0	2.248	1.0	0.028	1.0	0.125	3.0	0.040	1.0	1.60
	gmiGA-wrap	0.876	3.0	2.288	3.0	0.02826	4.0	0.125	4.0	0.041	3.0	3.40
	gmiLexGA-wrap	0.877	1.0	2.276	2.0	0.028	3.0	0.124	1.0	0.040	2.0	1.80
Art	gmiGA-filt	0.532	1.0	5.355	4.0	0.05822	3.0	0.585	1.0	0.148	3.0	2.40
	gmiLexGA-filt	0.532	2.0	5.247	1.0	0.058	1.0	0.591	4.0	0.144	1.0	1.80
	gmiGA-wrap	0.532	3.0	5.296	2.0	0.058	2.0	0.587	2.0	0.145	2.0	2.20
	gmiLexGA-wrap	0.529	4.0	5.354	3.0	0.058	4.0	0.589	3.0	0.148	4.0	3.60
Education	gmiGA-filt	0.561	1.0	3.826	2.0	0.0404	1.0	0.575	1.0	0.089	1.0	1.20
	gmiLexGA-filt	0.546	4.0	3.893	4.0	0.041	4.0	0.598	4.0	0.092	4.0	4.00
	gmiGA-wrap	0.555	2.0	3.818	1.0	0.04068	2.0	0.585	2.0	0.089	2.0	1.80
	gmiLexGA-wrap	0.548	3.0	3.880	3.0	0.041	3.0	0.594	3.0	0.091	3.0	3.00
Recreation	gmiGA-filt	0.378	4.0	5.703	4.0	0.065	4.0	0.805	4.0	0.220	4.0	4.00
	gmiLexGA-filt	0.575	3.0	4.134	3.0	0.055	2.0	0.543	3.0	0.150	3.0	2.80
	gmiGA-wrap	0.583	1.0	4.067	1.0	0.0546	1.0	0.533	1.0	0.147	1.0	1.00
	gmiLexGA-wrap	0.577	2.0	4.108	2.0	0.055	3.0	0.541	2.0	0.149	2.0	2.20
Health	gmiGA-filt	0.719	1.0	3.186	1.0	0.0397	1.0	0.352	1.0	0.058	1.0	1.00
	gmiLexGA-filt	0.701	4.0	3.268	4.0	0.042	4.0	0.377	4.0	0.060	4.0	4.00
	gmiGA-wrap	0.714	2.0	3.204	2.0	0.04056	2.0	0.356	2.0	0.058	2.0	2.00
	gmiLexGA-wrap	0.704	3.0	3.250	3.0	0.042	3.0	0.372	3.0	0.060	3.0	3.00
Ent.ment	gmiGA-filt	0.634	2.0	2.915	1.0	0.05334	1.0	0.483	1.0	0.105	1.0	1.20
	gmiLexGA-filt	0.623	3.0	3.011	3.0	0.055	3.0	0.497	3.0	0.109	3.0	3.00
	gmiGA-wrap	0.636	1.0	2.915	2.0	0.0539	2.0	0.484	2.0	0.105	2.0	1.80
	gmiLexGA-wrap	0.619	4.0	3.013	4.0	0.055	4.0	0.508	4.0	0.110	4.0	4.00
Computer	gmiGA-filt	0.649	1.0	4.124	2.0	0.03726	1.0	0.425	1.0	0.087	1.0	1.20
	gmiLexGA-filt	0.647	4.0	4.139	3.0	0.037	2.0	0.429	3.0	0.087	4.0	3.20
	gmiGA-wrap	0.647	3.0	4.108	1.0	0.03738	3.0	0.431	4.0	0.087	2.0	2.60
	gmiLexGA-wrap	0.648	2.0	4.143	4.0	0.038	4.0	0.427	2.0	0.087	3.0	3.00
Science	gmiGA-filt	0.486	1.0	6.707	1.0	0.0338	1.0	0.628	1.0	0.131	2.0	1.20
	gmiLexGA-filt	0.472	4.0	6.780	4.0	0.034	3.0	0.653	3.0	0.133	4.0	3.60
	gmiGA-wrap	0.484	2.0	6.716	2.0	0.03402	2.0	0.635	2.0	0.130	1.0	1.80
	gmiLexGA-wrap	0.472	3.0	6.758	3.0	0.034	4.0	0.654	4.0	0.132	3.0	3.40
MEAN	gmiGA-filt		1.9		2.4		1.9		1.6		2.2	2.00
	gmiLexGA-filt		3.1		2.8		2.6		3.2		2.9	2.92
	gmiGA-wrap		2.3		1.7		2.3		2.5		1.8	2.12
	gmiLexGA-wrap		2.7		3.1		3.2		2.7		3.1	2.96

Table 6.14: Summary of average ranking (AR) and the number of selected features (Sel.F) for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) when using ML-kNN as the classifier

ind.length	**gmiGA-filt**			**gmiLexGA-filt**			**gmiGA-wrap**			**gmiLexGA-wrap**		
	Sel.F	**%**	**AR**	**Sel.F**	**%**	**AR**	**Sel.F**	**%**	**AR**	**Sel.F**	**%**	**AR**
100	26.60	26.60	2.57	25.34	25.34	2.70	26.90	26.90	**2.33**	25.60	25.60	2.40
200	39.44	19.72	2.32	50.68	25.34	2.91	46.74	23.37	**2.08**	52.26	26.13	2.69
300	54.44	18.15	**1.86**	82.64	27.55	3.28	70.74	23.58	1.90	83.60	27.87	2.96
400	73.38	18.35	**2.00**	116.78	29.20	2.92	101.48	25.37	2.12	118.86	29.72	2.96
Overall	48.47	20.70	2.19	68.86	26.86	2.95	61.47	24.81	**2.11**	70.08	27.33	2.75

Table 6.15: Summary of overall average ranking (AR) across four individual lengths for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) when using ML-kNN as the classifier

Dataset	**Overall Average Rank (AR) across 4 individual lengths**			
	gmiGA-filt	**gmiLexGA-filt**	**gmiGA-wrap**	**gmiLexGA-wrap**
Enron	3.15(4)	2.50(3)	2.30(2)	2.05(1)
Medical	1.15(1)	3.35(4)	2.25(2)	3.25(3)
Business	2.83(4)	2.25(1)	2.63(3)	2.30(2)
Art	2.40(2)	2.95(4)	2.10(1)	2.55(3)
Education	1.55(1)	3.65(4)	2.00(2)	2.80(3)
Recreation	2.30(2)	3.40(4)	1.40(1)	2.90(3)
Health	2.45(2)	3.25(4)	1.8(1)	2.50(3)
Ent.ment	2.15(2)	3.10(3)	1.48(1)	3.28(4)
Computer	1.95(1)	3.40(4)	2.10(2)	2.55(3)
Science	1.70(1)	2.88(3)	2.38(2)	3.05(4)
Average	2.16(2)	3.07(3.4)	**2.04(1.7)**	2.72(2.9)

although gmiGA-wrap outperforms the other methods in terms of minimizing average ranking, gmiGA-filt obtained a better trade-off between minimizing average ranking and minimizing the number of selected features. That is, gmiGA-filt is just slightly worse than gmiGA-wrap in term of average ranking, but gmiGA-filt is substantially better than gmiGA-wrap in terms of number of selected features.

The overall average rank of each version of GA-ML-CFS for each dataset (averaged across the 4 GA individual lengths) is shown in Table 6.15. The first value in each cell is the actuall average rank, whilst the value between brackets is the

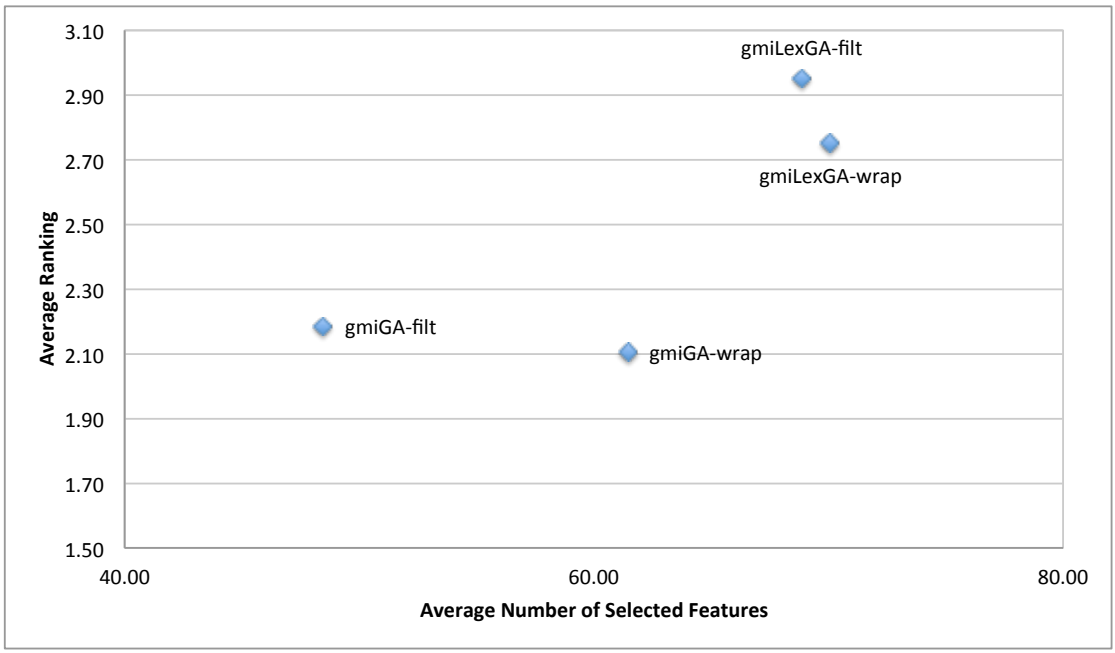


Figure 6.1: Overall average ranking (AR) for four versions of GA-ML-CFS plotted against the average number of selected features across all datasets and feature space sizes, when using ML-kNN as the classifier

"rank of the average rank". This later value was used in the Friedman and Holm's test. The Friedman test is suitable for comparing multiple algorithms on multiple domains (datasets). The null hypothesis is that there is no statistically significant difference between the classifiers' performance. If this null hypothesis is rejected, the Holm's posthoc test is used to identify which classifier has a predictive performance significantly different from the others [52].

For results in Table 6.15, there is a significant difference among the four GA-ML-CFS algorithms across the 10 evaluation datasets at the 0.05 level of significance for a two tailed test (p value = 0.00514). After we run the Holm's posthoc test on those results using gmiGA-wrap (the best method) as the control method, there is a significant difference between gmiGA-wrap and gmiLexGA-filt at the 0.05 level of significance (p value = 0.01941) but there is no significant difference between gmiGA-wrap and the other two algorithms.

6.7.2 BPMLL's Results for GA-ML-CFS and LexGA-Ml-CFS Using Mutual Information for Class Label Weighting with two parameter optimization approaches: wrapper-like approach versus filter approach

The results are shown in Tables 6.16 through 6.19. Recall that in these tables gmiGA-wrap denotes gmiGA-ML-CFS with parameter setting optimized by the wrapper-like approach (PS14 in Table 6.3) and gmiGA-filt denotes gmiGA-ML-CFS with parameters optimized by the filter approach (PS07 in Table 6.3). For LexGA versions, gmiLexGA-wrap denotes gmiLexGA-ML-CFS with parameter setting soptimized by the wrapper-like approach (PS02 in Table 6.3) and gmiLexGA-filt denotes gmiLexGA-ML-CFS with parameters optimized by the filter approach (PS10 in Table 6.3).

All GA results are an average over 5 runs with a different random seed used to create the initial population in each run. In Tables 6.16 - 6.19, the meaning of the columns are as explained in the beginning of Subsection 6.7.1.

When the individual length is 100 (Table 6.16), gmiGA-wrap obtained better predictive accuracy (lower average rank) than gmiGA-filt, gmiLexGA-wrap; and gmiLexGA-filt with 2.11 average rank versus 2.92, 2.78 and 2.19, respectively.

When the individual length is 200 (Table 6.17), gmiGA-wrap outperformed other versions of GA-ML-CFS with the smallest average rank (1.99); while gmiGA-filt, gmiLexGA-wrap and gmiLexGA-filt obtain a larger average rank of 2.60, 2.86 and 2.55, respectively.

When the individual length is 300 (Table 6.18), gmiLexGA-wrap obtained better predictive accuracy (lower average rank) than gmiLexGA-filt, gmiGA-wrap;

and gmiGA-filt, with average rank of 2.09 versus 2.39, 2.76 and 2.76, respectively.

Moreover, when the individual length is 400 (Table 6.19), gmiGA-filt outperformed other versions of GA-ML-CFS with the smallest average rank (1.82); while gmiGA-wrap, gmiLexGA-wrap; and gmiLexGA-filt obtain a larger average rank of 2.50, 2.68 and 3.00, respectively.

Table 6.20 shows the number and percentage of selected features (out of all input features) and average rank over 10 datasets for each individual length. Clearly, when the size of individual length equals to 100 and 200, gmiGA-wrap obtained the best overall average rank (2.11 and 1.99) with 26.94 % and 24.29 % of selected features. When the individual length is 300 gmiLexGA-wrap obtained the best predictive accuracy with 2.09 overall average rank, with 26.79% of selected features. When the individual length is 400, gmiGA-filt obtained the smallest overall average rank (1.82) and the smallest percentage (18.35%) of selected features.

Overall (last row of Table 6.20), gmiGA-wrap obtained the best average rank (2.34) and the second smallest percentage of selected features (25.73%); similarity to the results with ML-kNN in the previous subsection. Hence, gmiGA-wrap was the best version of GA-ML-CFS in both these experiments (with ML-kNN and with BPMLL), since maximizing accuracy is more important than minimizing the number of selected features.

Figure 6.2 shows the overall average ranking (AR) for four versions of GA-ML-CFS investigated in this section plotted against the average number of selected features across all datasets and feature space sizes, when using BPMLL as the classifier. Again, gmiGA-wrap outperforms the others in terms of predictive accuracy. However, gmiGA-filt outperforms the other methods in terms of the number of selected features.

Table 6.16: Predictive accuracy for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) with BPMLL Classifier (individual length = 100)

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiGA-filt	0.568	3.0	13.222	2.0	0.089	3.0	0.399	3.0	0.097	3.0	2.80
	gmiLexGA-filt	0.571	1.0	13.222	3.0	0.088	2.0	0.394	1.0	0.097	2.0	1.80
	gmiGA-wrap	0.568	4.0	13.259	4.0	0.088	1.0	0.405	4.0	0.097	4.0	3.40
	gmiLexGA-wrap	0.570	2.0	13.192	1.0	0.090	4.0	0.398	2.0	0.097	1.0	2.00
Medical	gmiGA-filt	0.694	3.0	2.693	2.0	0.030	2.0	0.453	4.0	0.044	2.0	2.60
	gmiLexGA-filt	0.696	2.0	2.727	4.0	0.030	3.0	0.443	2.0	0.046	4.0	3.00
	gmiGA-wrap	0.715	1.0	2.579	1.0	0.028	1.0	0.420	1.0	0.042	1.0	1.00
	gmiLexGA-wrap	0.693	4.0	2.724	3.0	0.031	4.0	0.448	3.0	0.045	3.0	3.40
Business	gmiGA-filt	0.852	2.0	2.766	1.0	0.042	1.0	0.139	2.0	0.049	1.0	1.40
	gmiLexGA-filt	0.851	4.0	2.788	4.0	0.043	3.0	0.139	1.0	0.049	4.0	3.20
	gmiGA-wrap	0.852	1.0	2.768	2.0	0.043	4.0	0.139	4.0	0.049	2.0	2.60
	gmiLexGA-wrap	0.852	3.0	2.778	3.0	0.042	2.0	0.139	3.0	0.049	3.0	2.80
Art	gmiGA-filt	0.436	3.5	6.006	3.5	0.186	4.0	0.752	3.0	0.175	3.5	3.50
	gmiLexGA-filt	0.437	1.0	5.978	1.0	0.181	1.0	0.752	1.0	0.174	1.0	1.00
	gmiGA-wrap	0.436	2.0	6.000	2.0	0.184	2.0	0.752	3.0	0.175	2.0	2.20
	gmiLexGA-wrap	0.436	3.5	6.006	3.5	0.185	3.0	0.752	3.0	0.175	3.5	3.30
Education	gmiGA-filt	0.478	4.0	4.608	4.0	0.126	4.0	0.681	4.0	0.109	4.0	4.00
	gmiLexGA-filt	0.479	2.0	4.587	3.0	0.120	2.0	0.680	2.0	0.109	3.0	2.40
	gmiGA-wrap	0.479	1.0	4.560	1.0	0.120	1.0	0.680	1.0	0.108	1.0	1.00
	gmiLexGA-wrap	0.478	3.0	4.585	2.0	0.123	3.0	0.681	3.0	0.109	2.0	2.60
Recreation	gmiGA-filt	0.383	2.0	5.403	2.0	0.197	4.0	0.798	2.0	0.216	2.0	2.40
	gmiLexGA-filt	0.378	3.0	5.423	3.0	0.193	2.0	0.806	4.0	0.217	3.0	3.00
	gmiGA-wrap	0.387	1.0	5.357	1.0	0.193	3.0	0.794	1.0	0.213	1.0	1.40
	gmiLexGA-wrap	0.377	4.0	5.442	4.0	0.188	1.0	0.804	3.0	0.218	4.0	3.20
Health	gmiGA-filt	0.621	4.0	3.933	1.0	0.110	2.0	0.486	4.0	0.077	1.0	2.40
	gmiLexGA-filt	0.621	1.0	3.938	4.0	0.113	4.0	0.485	1.0	0.077	4.0	2.80
	gmiGA-wrap	0.621	3.0	3.935	2.0	0.111	3.0	0.485	3.0	0.077	2.0	2.60
	gmiLexGA-wrap	0.621	2.0	3.936	3.0	0.109	1.0	0.485	2.0	0.077	3.0	2.20
Ent.ment	gmiGA-filt	0.528	4.0	3.470	4.0	0.153	2.0	0.649	3.0	0.132	4.0	3.40
	gmiLexGA-filt	0.529	1.0	3.460	1.0	0.152	1.0	0.649	2.0	0.132	1.0	1.20
	gmiGA-wrap	0.528	3.0	3.467	3.0	0.154	3.0	0.649	1.0	0.132	3.0	2.60
	gmiLexGA-wrap	0.529	2.0	3.461	2.0	0.156	4.0	0.649	4.0	0.132	2.0	2.80
Computer	gmiGA-filt	0.598	4.0	4.906	4.0	0.078	1.0	0.475	3.0	0.103	4.0	3.20
	gmiLexGA-filt	0.598	2.0	4.891	2.0	0.079	2.0	0.475	3.0	0.103	3.0	2.40
	gmiGA-wrap	0.599	1.0	4.866	1.0	0.082	4.0	0.475	3.0	0.102	1.0	2.00
	gmiLexGA-wrap	0.598	3.0	4.900	3.0	0.079	3.0	0.475	1.0	0.102	2.0	2.40
Science	gmiGA-filt	0.395	4.0	7.872	4.0	0.132	4.0	0.758	1.5	0.158	4.0	3.50
	gmiLexGA-filt	0.396	1.0	7.797	1.0	0.128	1.0	0.758	1.5	0.156	1.0	1.10
	gmiGA-wrap	0.396	2.0	7.842	2.0	0.129	2.0	0.758	3.5	0.157	2.0	2.30
	gmiLexGA-wrap	0.395	3.0	7.863	3.0	0.132	3.0	0.758	3.5	0.157	3.0	3.10
MEAN	gmiGA-filt		3.4		2.8		2.7		3.0		2.9	2.92
	gmiLexGA-filt		1.8		2.6		2.1		1.9		2.6	2.19
	gmiGA-wrap		1.9		1.9		2.4		2.5		1.9	2.11
	gmiLexGA-wrap		3.0		2.8		2.8		2.8		2.7	2.78

Table 6.17: Predictive accuracy for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) with BPMLL Classifier (individual length = 200)

Dataset	**Methods**	**Predictive Accuracy Measures and Ranking**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiGA-filt	0.564	2.0	13.300	1.0	0.088	1.0	0.390	1.0	0.098	2.0	1.40
	gmiLexGA-filt	0.559	3.0	13.420	3.0	0.089	4.0	0.422	4.0	0.099	3.0	3.40
	gmiGA-wrap	0.565	1.0	13.333	2.0	0.088	2.0	0.401	2.0	0.098	1.0	1.60
	gmiLexGA-wrap	0.558	4.0	13.721	4.0	0.089	3.0	0.411	3.0	0.101	4.0	3.60
Medical	gmiGA-filt	0.774	4.0	2.498	4.0	0.023	4.0	0.315	4.0	0.039	4.0	4.00
	gmiLexGA-filt	0.792	3.0	2.430	3.0	0.021	3.0	0.286	3.0	0.038	3.0	3.00
	gmiGA-wrap	0.815	1.0	2.282	1.0	0.018	1.0	0.256	1.0	0.035	1.0	1.00
	gmiLexGA-wrap	0.794	2.0	2.410	2.0	0.020	2.0	0.282	2.0	0.037	2.0	2.00
Business	gmiGA-filt	0.853	3.0	2.725	3.0	0.043	3.0	0.139	2.5	0.048	3.0	2.90
	gmiLexGA-filt	0.853	4.0	2.745	4.0	0.042	2.0	0.139	2.5	0.048	4.0	3.30
	gmiGA-wrap	0.854	1.0	2.715	1.0	0.043	4.0	0.139	2.5	0.048	1.0	1.90
	gmiLexGA-wrap	0.854	2.0	2.721	2.0	0.042	1.0	0.139	2.5	0.048	2.0	1.90
Art	gmiGA-filt	0.436	2.0	5.999	4.0	0.191	2.0	0.752	2.5	0.175	3.0	2.70
	gmiLexGA-filt	0.436	3.0	5.991	1.0	0.191	1.0	0.752	2.5	0.175	2.0	1.90
	gmiGA-wrap	0.437	1.0	5.994	2.0	0.192	3.0	0.752	2.5	0.175	1.0	1.90
	gmiLexGA-wrap	0.436	4.0	5.995	3.0	0.197	4.0	0.752	2.5	0.175	4.0	3.50
Education	gmiGA-filt	0.481	2.0	4.504	1.0	0.130	4.0	0.679	1.0	0.107	1.0	1.80
	gmiLexGA-filt	0.479	4.0	4.598	4.0	0.119	1.0	0.681	4.0	0.109	4.0	3.40
	gmiGA-wrap	0.481	1.0	4.505	2.0	0.124	3.0	0.681	2.0	0.107	2.0	2.00
	gmiLexGA-wrap	0.481	3.0	4.542	3.0	0.120	2.0	0.681	3.0	0.108	3.0	2.80
Recreation	gmiGA-filt	0.378	3.0	5.563	4.0	0.215	1.0	0.804	1.0	0.220	3.0	2.40
	gmiLexGA-filt	0.377	4.0	5.561	3.0	0.234	3.0	0.805	4.0	0.220	4.0	3.60
	gmiGA-wrap	0.379	1.0	5.517	1.0	0.225	2.0	0.804	2.0	0.219	1.0	1.40
	gmiLexGA-wrap	0.379	2.0	5.531	2.0	0.237	4.0	0.804	3.0	0.219	2.0	2.60
Health	gmiGA-filt	0.614	4.0	3.999	4.0	0.103	1.0	0.489	3.0	0.079	4.0	3.20
	gmiLexGA-filt	0.616	2.0	3.893	1.0	0.116	4.0	0.489	2.0	0.075	1.0	2.00
	gmiGA-wrap	0.616	3.0	3.934	3.0	0.111	2.0	0.488	1.0	0.077	3.0	2.40
	gmiLexGA-wrap	0.616	1.0	3.895	2.0	0.115	3.0	0.489	4.0	0.076	2.0	2.40
Ent.ment	gmiGA-filt	0.520	3.0	3.513	3.0	0.162	1.0	0.662	3.0	0.135	3.0	2.60
	gmiLexGA-filt	0.522	1.0	3.498	1.0	0.167	2.0	0.660	1.0	0.134	1.0	1.20
	gmiGA-wrap	0.520	2.0	3.521	4.0	0.170	3.0	0.662	2.0	0.135	4.0	3.00
	gmiLexGA-wrap	0.519	4.0	3.502	2.0	0.174	4.0	0.667	4.0	0.134	2.0	3.20
Computer	gmiGA-filt	0.597	4.0	4.883	3.0	0.084	1.0	0.475	2.5	0.103	3.0	2.70
	gmiLexGA-filt	0.599	1.0	4.831	1.0	0.086	3.0	0.475	1.0	0.102	1.0	1.40
	gmiGA-wrap	0.599	2.0	4.864	2.0	0.086	4.0	0.475	4.0	0.102	2.0	2.80
	gmiLexGA-wrap	0.598	3.0	4.884	4.0	0.086	2.0	0.475	2.5	0.103	4.0	3.10
Science	gmiGA-filt	0.397	3.0	7.789	3.0	0.130	1.0	0.758	2.5	0.156	2.0	2.30
	gmiLexGA-filt	0.397	2.0	7.771	2.0	0.134	2.0	0.758	2.5	0.156	3.0	2.30
	gmiGA-wrap	0.397	1.0	7.741	1.0	0.136	4.0	0.758	2.5	0.156	1.0	1.90
	gmiLexGA-wrap	0.396	4.0	7.792	4.0	0.135	3.0	0.758	2.5	0.157	4.0	3.50
MEAN	gmiGA-filt		3.0		3.0		1.9		2.3		2.8	2.60
	gmiLexGA-filt		2.7		2.3		2.5		2.7		2.6	2.55
	gmiGA-wrap		1.4		1.9		2.8		2.2		1.7	1.99
	gmiLexGA-wrap		2.9		2.8		2.8		2.9		2.9	2.86

Table 6.18: Predictive accuracy for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) with BPMLL Classifier (individual length = 300)

Dataset	**Methods**	**BPMLL Classifier**									
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiGA-filt	0.568	1.0	13.253	1.0	0.088	1.0	0.395	1.0	0.097	1.0	1.00
	gmiLexGA-filt	0.568	2.0	14.012	4.0	0.092	4.0	0.421	4.0	0.102	4.0	3.60
	gmiGA-wrap	0.563	3.0	13.640	2.0	0.091	3.0	0.415	2.0	0.100	2.0	2.40
	gmiLexGA-wrap	0.562	4.0	13.750	3.0	0.091	2.0	0.419	3.0	0.100	3.0	3.00
Medical	gmiGA-filt	0.796	4.0	2.344	3.0	0.021	4.0	0.282	4.0	0.036	4.0	3.80
	gmiLexGA-filt	0.830	1.0	2.354	4.0	0.016	1.0	0.221	1.0	0.036	3.0	2.00
	gmiGA-wrap	0.825	2.0	2.272	2.0	0.017	3.0	0.235	2.0	0.034	2.0	2.20
	gmiLexGA-wrap	0.821	3.0	2.209	1.0	0.017	2.0	0.244	3.0	0.033	1.0	2.00
Business	gmiGA-filt	0.842	4.0	2.841	4.0	0.052	4.0	0.139	2.5	0.052	4.0	3.70
	gmiLexGA-filt	0.856	1.0	2.685	1.0	0.042	3.0	0.139	2.5	0.047	1.0	1.70
	gmiGA-wrap	0.853	3.0	2.762	3.0	0.034	1.0	0.139	2.5	0.049	3.0	2.50
	gmiLexGA-wrap	0.854	2.0	2.719	2.0	0.042	2.0	0.139	2.5	0.048	2.0	2.10
Art	gmiGA-filt	0.521	1.0	5.256	1.0	0.060	1.0	0.607	1.0	0.144	1.0	1.00
	gmiLexGA-filt	0.436	3.0	5.994	3.0	0.205	2.0	0.752	2.5	0.175	3.0	2.70
	gmiGA-wrap	0.167	4.0	10.195	4.0	0.626	4.0	0.973	4.0	0.347	4.0	4.00
	gmiLexGA-wrap	0.437	2.0	5.992	2.0	0.207	3.0	0.752	2.5	0.175	2.0	2.30
Education	gmiGA-filt	0.472	4.0	4.983	4.0	0.169	4.0	0.681	3.5	0.117	4.0	3.90
	gmiLexGA-filt	0.480	3.0	4.582	3.0	0.141	3.0	0.681	2.0	0.108	3.0	2.80
	gmiGA-wrap	0.541	1.0	3.914	1.0	0.041	1.0	0.604	1.0	0.092	1.0	1.00
	gmiLexGA-wrap	0.481	2.0	4.555	2.0	0.139	2.0	0.681	3.5	0.108	2.0	2.30
Recreation	gmiGA-filt	0.375	3.0	5.682	2.0	0.236	1.0	0.805	3.5	0.224	2.0	2.30
	gmiLexGA-filt	0.376	2.0	5.687	3.0	0.321	3.0	0.804	1.0	0.225	3.0	2.40
	gmiGA-wrap	0.370	4.0	6.133	4.0	0.481	4.0	0.805	3.5	0.242	4.0	3.90
	gmiLexGA-wrap	0.378	1.0	5.632	1.0	0.293	2.0	0.804	2.0	0.223	1.0	1.40
Health	gmiGA-filt	0.491	4.0	3.696	1.0	0.235	4.0	0.699	4.0	0.145	4.0	3.40
	gmiLexGA-filt	0.612	2.0	3.880	2.0	0.120	1.0	0.489	1.0	0.075	1.0	1.40
	gmiGA-wrap	0.607	3.0	4.094	4.0	0.132	3.0	0.489	3.0	0.081	3.0	3.20
	gmiLexGA-wrap	0.612	1.0	3.883	3.0	0.123	2.0	0.489	2.0	0.075	2.0	2.00
Ent.ment	gmiGA-filt	0.598	1.0	4.911	3.0	0.087	1.0	0.475	1.5	0.104	1.0	1.50
	gmiLexGA-filt	0.474	4.0	3.848	2.0	0.263	4.0	0.717	4.0	0.151	4.0	3.60
	gmiGA-wrap	0.598	2.0	4.914	4.0	0.091	2.0	0.475	1.5	0.104	2.0	2.30
	gmiLexGA-wrap	0.484	3.0	3.773	1.0	0.241	3.0	0.705	3.0	0.148	3.0	2.60
Computer	gmiGA-filt	0.396	4.0	7.815	4.0	0.129	3.0	0.758	3.5	0.157	4.0	3.70
	gmiLexGA-filt	0.595	2.0	4.935	2.0	0.093	1.0	0.475	1.5	0.104	2.0	1.70
	gmiGA-wrap	0.396	3.0	7.790	3.0	0.133	4.0	0.758	3.5	0.157	3.0	3.30
	gmiLexGA-wrap	0.596	1.0	4.926	1.0	0.094	2.0	0.475	1.5	0.104	1.0	1.30
Science	gmiGA-filt	0.395	3.0	7.984	3.0	0.184	4.0	0.758	2.5	0.161	4.0	3.30
	gmiLexGA-filt	0.395	1.5	7.893	2.0	0.162	2.0	0.758	2.5	0.159	2.0	2.00
	gmiGA-wrap	0.395	1.5	7.987	4.0	0.182	3.0	0.758	2.5	0.161	3.0	2.80
	gmiLexGA-wrap	0.395	4.0	7.864	1.0	0.159	1.0	0.758	2.5	0.159	1.0	1.90
MEAN	gmiGA-filt		2.9		2.6		2.7		2.7		2.9	2.76
	gmiLexGA-filt		2.2		2.6		2.4		2.2		2.6	2.39
	gmiGA-wrap		2.7		3.1		2.8		2.6		2.7	2.76
	gmiLexGA-wrap		2.3		1.7		2.1		2.6		1.8	2.09

Table 6.19: Predictive accuracy for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) with BPMLL Classifier (individual length = 400)

Dataset	**Methods**	**BPMLL Classifier**										
		Avg-Pre	**R**	**Coverage**	**R**	**H-Loss**	**R**	**OneError**	**R**	**R-Loss**	**R**	**AR**
Enron	gmiGA-filt	0.567	1.0	13.380	1.0	0.088	1.0	0.405	1.0	0.098	1.0	1.00
	gmiLexGA-filt	0.563	3.0	14.320	3.0	0.093	3.0	0.427	3.0	0.105	3.0	3.00
	gmiGA-wrap	0.566	2.0	13.995	2.0	0.092	2.0	0.428	4.0	0.102	2.0	2.40
	gmiLexGA-wrap	0.562	4.0	14.606	4.0	0.095	4.0	0.421	2.0	0.107	4.0	3.60
Medical	gmiGA-filt	0.808	4.0	2.387	4.0	0.018	4.0	0.263	3.0	0.037	4.0	3.80
	gmiLexGA-filt	0.819	2.0	2.334	3.0	0.017	3.0	0.242	2.0	0.036	3.0	2.60
	gmiGA-wrap	0.809	3.0	2.321	2.0	0.017	2.0	0.267	4.0	0.035	2.0	2.60
	gmiLexGA-wrap	0.835	1.0	2.201	1.0	0.016	1.0	0.215	1.0	0.033	1.0	1.00
Business	gmiGA-filt	0.856	4.0	2.677	4.0	0.038	1.0	0.139	1.0	0.047	4.0	2.80
	gmiLexGA-filt	0.856	3.0	2.657	3.0	0.042	2.0	0.139	3.0	0.047	3.0	2.80
	gmiGA-wrap	0.858	1.0	2.630	1.0	0.042	4.0	0.139	3.0	0.046	1.0	2.00
	gmiLexGA-wrap	0.857	2.0	2.641	2.0	0.042	3.0	0.139	3.0	0.046	2.0	2.40
Art	gmiGA-filt	0.436	1.0	5.999	1.0	0.207	1.0	0.752	2.0	0.175	1.0	1.20
	gmiLexGA-filt	0.434	3.0	6.139	3.0	0.284	3.0	0.752	4.0	0.180	3.0	3.20
	gmiGA-wrap	0.432	4.0	6.209	4.0	0.319	4.0	0.752	2.0	0.183	4.0	3.60
	gmiLexGA-wrap	0.436	2.0	6.045	2.0	0.260	2.0	0.752	2.0	0.177	2.0	2.00
Education	gmiGA-filt	0.480	1.0	4.560	1.0	0.122	1.0	0.681	2.5	0.108	1.0	1.30
	gmiLexGA-filt	0.476	4.0	4.683	4.0	0.148	4.0	0.681	2.5	0.111	4.0	3.70
	gmiGA-wrap	0.479	2.0	4.616	2.0	0.140	2.0	0.681	2.5	0.109	2.0	2.10
	gmiLexGA-wrap	0.478	3.0	4.657	3.0	0.146	3.0	0.681	2.5	0.110	3.0	2.90
Recreation	gmiGA-filt	0.372	4.0	5.885	2.0	0.376	2.0	0.805	2.5	0.233	2.0	2.50
	gmiLexGA-filt	0.373	3.0	5.906	3.0	0.388	3.0	0.805	2.5	0.235	4.0	3.10
	gmiGA-wrap	0.374	1.0	5.831	1.0	0.368	1.0	0.805	2.5	0.232	1.0	1.30
	gmiLexGA-wrap	0.373	2.0	5.920	4.0	0.402	4.0	0.805	2.5	0.235	3.0	3.10
Health	gmiGA-filt	0.619	1.0	3.818	2.0	0.114	1.0	0.489	4.0	0.073	2.0	2.00
	gmiLexGA-filt	0.613	3.0	3.914	3.0	0.141	3.0	0.489	1.0	0.076	3.0	2.60
	gmiGA-wrap	0.618	2.0	3.814	1.0	0.118	2.0	0.489	2.0	0.073	1.0	1.60
	gmiLexGA-wrap	0.609	4.0	3.965	4.0	0.142	4.0	0.489	3.0	0.077	4.0	3.80
Ent.ment	gmiGA-filt	0.476	1.0	3.857	1.0	0.262	1.0	0.712	1.0	0.152	1.0	1.00
	gmiLexGA-filt	0.467	4.0	3.986	3.0	0.283	3.0	0.722	4.0	0.157	3.0	3.40
	gmiGA-wrap	0.469	2.0	4.021	4.0	0.301	4.0	0.715	2.0	0.157	4.0	3.20
	gmiLexGA-wrap	0.467	3.0	3.953	2.0	0.276	2.0	0.722	3.0	0.155	2.0	2.40
Computer	gmiGA-filt	0.595	1.0	4.954	1.0	0.091	1.0	0.475	2.5	0.104	1.0	1.30
	gmiLexGA-filt	0.580	3.0	5.097	2.0	0.128	4.0	0.475	2.5	0.109	2.0	2.70
	gmiGA-wrap	0.583	2.0	5.131	4.0	0.126	3.0	0.475	2.5	0.110	3.0	2.90
	gmiLexGA-wrap	0.580	4.0	5.119	3.0	0.123	2.0	0.475	2.5	0.110	4.0	3.10
Science	gmiGA-filt	0.395	1.0	7.886	1.0	0.154	1.0	0.758	2.5	0.159	1.0	1.30
	gmiLexGA-filt	0.393	3.0	8.109	2.0	0.246	4.0	0.758	2.5	0.165	3.0	2.90
	gmiGA-wrap	0.393	4.0	8.133	4.0	0.235	2.0	0.758	2.5	0.165	4.0	3.30
	gmiLexGA-wrap	0.393	2.0	8.117	3.0	0.246	3.0	0.758	2.5	0.164	2.0	2.50
MEAN	gmiGA-filt		1.9		1.8		1.4		2.2		1.8	1.82
	gmiLexGA-filt		3.1		2.9		3.2		2.7		3.1	3.00
	gmiGA-wrap		2.3		2.5		2.6		2.7		2.4	2.50
	gmiLexGA-wrap		2.7		2.8		2.8		2.4		2.7	2.68

Table 6.20: Summary of average ranking (AR) and the number and percentage of selected features (Sel.F) for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) when using BPMLL as the classifier

ind.length	**gmiGA-filt**			**gmiLexGA-filt**			**gmiGA-wrap**			**gmiLexGA-wrap**		
	Sel.F	**%**	**AR**	**Sel.F**	**%**	**AR**	**Sel.F**	**%**	**AR**	**Sel.F**	**%**	**AR**
100	26.60	26.60	2.92	25.34	25.34	2.19	26.94	26.94	**2.11**	25.76	25.76	2.78
200	39.44	19.72	2.60	50.68	25.34	2.55	48.58	24.29	**1.99**	50.74	25.37	2.86
300	54.44	18.15	2.76	82.64	27.55	2.39	73.96	24.65	2.76	80.90	26.97	**2.09**
400	73.38	18.35	**1.82**	116.78	29.20	3.00	108.12	27.03	2.50	116.50	29.13	2.68
Overall	48.47	20.70	2.53	68.86	26.86	2.53	64.40	25.73	**2.34**	68.48	26.81	2.60

Table 6.21: Summary of overall average ranking (AR) across four individual lengths for four versions of GA-ML-CFS using mutual information for class label weighting with two parameter optimization approaches: wrapper-like approach (gmiGA-wrap/gmiLexGA-wrap) versus filter-like approach (gmiGA-filt/gmiLexGA-filt) when using BPMLL as the classifier

Dataset	**Overall Average Rank (AR) across 4 individual lengths**			
	gmiGA-filt	**gmiLexGA-filt**	**gmiGA-wrap**	**gmiLexGA-wrap**
Enron	1.55(1)	2.95(3)	2.45(2)	3.05(4)
Medical	3.55(4)	2.65(3)	1.70(1)	2.10(2)
Business	2.70(3)	2.75(4)	2.25(1)	2.30(2)
Art	2.10(1)	2.20(2)	2.93(4)	2.78(3)
Education	2.75(3)	3.08(4)	1.53(1)	2.65(2)
Recreation	2.40(2)	3.03(4)	2.00(1)	2.58(3)
Health	2.75(4)	2.20(1)	2.45(2)	2.60(3)
Ent.ment	2.13(1)	2.35(2)	2.78(4)	2.75(3)
Computer	2.73(3)	2.05(1)	2.75(4)	2.48(2)
Science	2.60(3)	2.08(1)	2.58(2)	2.75(4)
Average	2.53(2.5)	2.53(2.5)	**2.34(2.2)**	2.6(2.8)

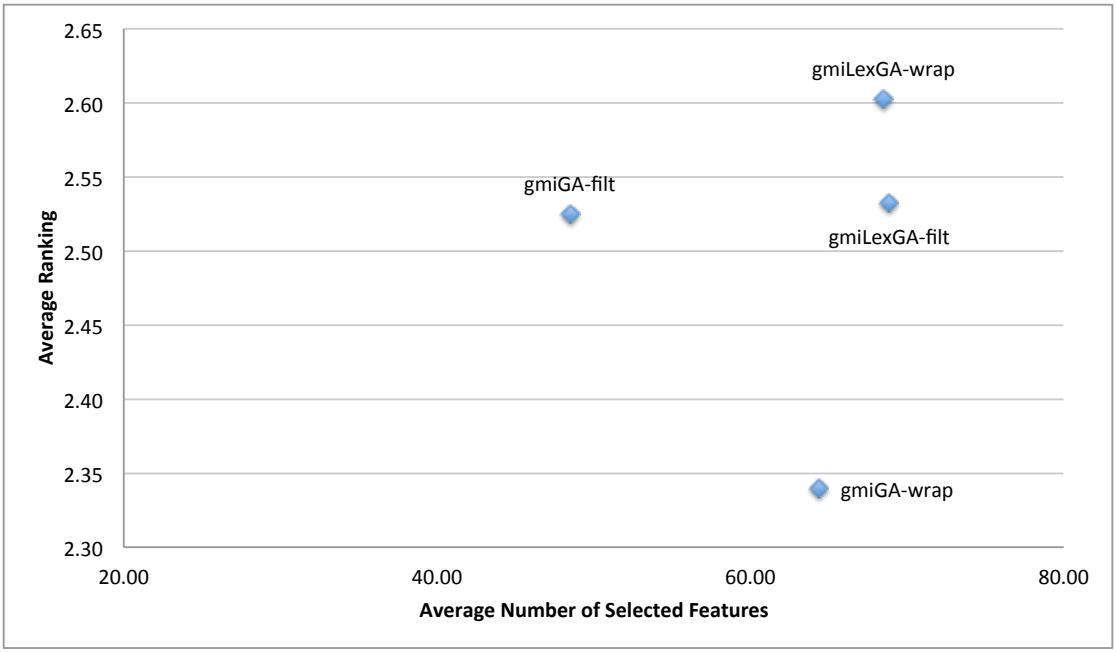


Figure 6.2: Overall average ranking (AR) for four versions of GA-ML-CFS plotted against the average number of selected features across all datasets and feature space sizes, when using BPMLL as the classifier

The overall average rank of each version of GA-ML-CFS for each dataset (averaged across the 4 GA individual lengths) is shown in Table 6.21. The first value in each cell is the actual average rank, whilst the value between brackets is the "rank of the average rank". This later value was used in the Friedman and Holm's test (as discussed at the end of Subsection 6.7.1). There are no significant differences among the four GA-ML-CFS algorithms across the 10 evaluation datasets, according to the Friedman test at the 0.05 significance level.

6.8 Results Comparing the Best Version of GA-ML-CFS (gmiGA-wrap) and Other Multi-Label Feature Selection Methods

6.8.1 Methods Being Compared and Experimental Methodology

In this Section we compare the best version of our GA-ML-CFS method according to the results reported in previous Section, namely gmiGA-wrap, with several other multi-label feature selection methods, namely ML-CFS using mutual information for class label weighting (gmiML-CFS), Relief for Multi-Label feature selection (RFML) and three different baseline approaches: Binary Relevance (BR), Correlation-Based Feature Selection with the union operator (CFS-U) and No feature selection (NoFS). Since the datasets have very large number of features (from 1,001 to 37,187), for all approaches, the univariate filter approach was used for all datasets, as described in Section 6.2.7, in order to reduce the feature space size and reduce computational time.

Recall that gmiGA-wrap is the GA-ML-CFS version using mutual information for class label weighting and absolute value of correlation coefficient, and with parameter setting optimized by the wrapper-like approach.

gmiML-CFS is the hill-climbing based ML-CFS proposed in Chapter 4. Recall that gmiML-CFS takes label dependences into account. We used mutual information (MI) to measure the degree of dependence between each pair of labels. The details of this approach are described in Section 4.2.2.

The RFML method is a well-known multi-label feature selection method proposed in [105], as discussed in Section 3.4. We used the RFML implementation

kindly provided by the authors; with default parameter setting. After running RFML and obtaining its feature ranking, we selected the top k features in the ranking, where k is the same number of features selected by gmiGA-wrap.

Binary Relevance (BR) is provided in the multi-label classification repository. This approach was discussed in Section 3.1 and it essentially consists of the base classifier (in our case kNN and multilayer perceptron which are provided on the Weka website) without any feature selection in a pre-processing step. Each base classifier was used with its default parameter setting.

The CFS-U approach consists of running a conventional single-label CFS method for selecting a feature subset for each class label separately and then returning the union of those selected feature subsets as the set of features to be given to the multi-label classification algorithm. The CFS implementation used in our experiments was the single-label CFSSubsetEval method in the well-known Weka data mining tool [44]. This method was used with its default parameters, and it evaluates candidate feature subsets according to Equation (4.1).

In the NoFS approach, we give all input features to the classifier. Recall that the NoFS and BR approaches still involve some initial feature selection based on the univariate approach, based on Equation (4.3), since that univariate approach was applied to all datasets in a pre-processing step, regardless of whether or not a feature selection method is applied.

Similarly to the previous Sections of results in this Chapter, in the next two Sections we report results separately for the experiments using ML-kNN and BPMLL as the classifier.

6.8.2 Results for the Best Version of GA-ML-CFS (gmiGA-wrap) and Other Multi-Label Feature Selection Methods using the ML-kNN Classifier

The results are shown in Tables 6.22 - 6.25 for feature space size varying from 100 to 400, respectively. The meanings of the table columns are as explained in the beginning of Section 6.7.1. In Table 6.22, when the feature space size equals to 100, CFS-U obtained the best overall average rank (2.25); while gmiGA-wrap obtained the second best overall average rank (2.66) and outperformed NoFS, gmiML-CFS, RFML and BR, which obtained overall average rank 2.79, 3.12, 4.64 and 5.54, respectively.

In Table 6.23, the best method was again CFS-U, with an overall average rank of 1.86. The second best method was again gmiGA-wrap with overall average rank = 2.52, which outperformed NoFS, RFML, gmiML-CFS, and BR which obtained 2.71, 3.93, 4.55 and 5.43 overall average rank, respectively.

In Table 6.24 the best method was gmiGA-wrap, with an overall average rank of 2.20. This method outperformed gmiML-CFS, CFS-U, NoFS, RFML and BR, which obtained overall average rank 2.35, 2.45, 3.60, 4.52 and 5.88, respectively.

In table 6.25 the best method was CFS-U with an overall average rank of 2.25; while the second best method was gmiGA-wrap (2.31), which outperformed NoFS, BR and RFML which obtained average ranks 2.86, 3.36, 4.22 and 6.00, respectively.

Table 6.26 reports the summary of results in terms of the overall average ranking and the number of selected features of gmiGA-wrap and multi-label feature selection approaches when using MLkNN as the classifier. In each table row, the Average Rank (AR) values are the same as reported in the last row of the Table for

Table 6.22: Values of five multi-label predictive accuracy measures for ML-kNN classifier with six different multi-label feature selection methods - feature space size = 100

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiGA-wrap	0.584	3.0	13.763	4.0	0.057	3.0	0.394	4.0	0.100	4.0	3.60
	gmi-ML-CFS	0.583	4.0	13.679	3.0	0.057	1.0	0.389	2.0	0.100	3.0	2.60
	NoFS	0.584	2.0	13.380	1.0	0.058	4.0	0.396	5.0	0.097	1.0	2.60
	BR(kNN)	0.547	6.0	14.109	6.0	0.098	6.0	0.413	6.0	0.106	6.0	6.00
	CFS-U	0.587	1.0	13.501	2.0	0.057	2.0	0.390	3.0	0.098	2.0	2.00
	RFML	0.581	5.0	13.883	5.0	0.059	5.0	0.385	1.0	0.103	5.0	4.20
Medical	gmiGA-wrap	0.793	2.0	3.111	2.0	0.015	1.0	0.256	1.0	0.050	2.0	1.60
	gmi-ML-CFS	0.760	4.0	3.372	4.0	0.017	3.0	0.301	4.5	0.055	4.0	3.90
	NoFS	0.717	6.0	3.614	6.0	0.019	6.0	0.374	6.0	0.062	6.0	6.00
	BR(kNN)	0.796	1.0	2.299	1.0	0.017	4.0	0.281	2.0	0.037	1.0	1.80
	CFS-U	0.758	5.0	3.505	5.0	0.018	5.0	0.301	4.5	0.059	5.0	4.90
	RFML	0.767	3.0	3.304	3.0	0.017	2.0	0.293	3.0	0.054	3.0	2.80
Business	gmiGA-wrap	0.874	2.0	2.386	4.0	0.028	2.0	0.124	3.0	0.043	4.0	3.00
	gmi-ML-CFS	0.874	3.0	2.371	2.0	0.028	4.0	0.123	2.0	0.043	2.5	2.70
	NoFS	0.874	4.0	2.369	1.0	0.028	3.0	0.124	4.0	0.043	1.0	2.60
	BR(kNN)	0.854	6.0	2.725	6.0	0.042	6.0	0.139	6.0	0.048	6.0	6.00
	CFS-U	0.875	1.0	2.379	3.0	0.028	1.0	0.122	1.0	0.043	2.5	1.70
	RFML	0.870	5.0	2.439	5.0	0.029	5.0	0.129	5.0	0.044	5.0	5.00
Art	gmiGA-wrap	0.527	4.0	5.409	4.0	0.059	2.0	0.589	3.0	0.150	4.0	3.40
	gmi-ML-CFS	0.528	3.0	5.398	3.0	0.059	1.0	0.588	2.0	0.150	3.0	2.40
	NoFS	0.529	2.0	5.306	2.0	0.059	4.0	0.592	4.0	0.146	2.0	2.80
	BR(kNN)	0.432	6.0	5.971	5.0	0.229	6.0	0.752	6.0	0.176	5.0	5.60
	CFS-U	0.533	1.0	5.272	1.0	0.059	3.0	0.586	1.0	0.145	1.0	1.40
	RFML	0.436	5.0	6.127	6.0	0.064	5.0	0.748	5.0	0.178	6.0	5.40
Education	gmiGA-wrap	0.544	2.0	3.924	2.0	0.042	3.0	0.600	2.0	0.092	1.0	2.00
	gmi-ML-CFS	0.480	5.0	4.532	5.0	0.134	5.0	0.679	5.0	0.107	5.0	5.00
	NoFS	0.543	3.0	3.938	3.0	0.041	1.5	0.602	3.0	0.093	3.0	2.70
	BR(kNN)	0.476	6.0	4.645	6.0	0.145	6.0	0.681	6.0	0.110	6.0	6.00
	CFS-U	0.545	1.0	3.921	1.0	0.041	1.5	0.597	1.0	0.092	2.0	1.30
	RFML	0.486	4.0	4.420	4.0	0.045	4.0	0.679	4.0	0.106	4.0	4.00
Recreation	gmiGA-wrap	0.536	2.0	4.286	1.0	0.059	2.0	0.603	4.0	0.157	1.0	2.00
	gmi-ML-CFS	0.535	3.0	4.349	4.0	0.059	3.0	0.601	3.0	0.159	4.0	3.40
	NoFS	0.536	1.0	4.333	3.0	0.058	1.0	0.595	1.0	0.157	3.0	1.80
	BR(kNN)	0.376	6.0	5.603	6.0	0.346	6.0	0.805	6.0	0.222	6.0	6.00
	CFS-U	0.535	4.0	4.302	2.0	0.059	4.0	0.598	2.0	0.157	2.0	2.80
	RFML	0.396	5.0	5.301	5.0	0.065	5.0	0.783	5.0	0.205	5.0	5.00
Health	gmiGA-wrap	0.631	4.0	3.787	4.0	0.049	4.0	0.478	4.0	0.075	2.0	3.60
	gmi-ML-CFS	0.634	1.0	3.747	1.0	0.049	3.0	0.476	2.0	0.075	4.0	2.20
	NoFS	0.631	3.0	3.784	3.0	0.049	1.5	0.476	1.0	0.075	3.0	2.30
	BR(kNN)	0.616	6.0	4.062	6.0	0.129	6.0	0.489	6.0	0.078	6.0	6.00
	CFS-U	0.632	2.0	3.767	2.0	0.049	1.5	0.477	3.0	0.075	1.0	1.90
	RFML	0.625	5.0	3.905	5.0	0.050	5.0	0.482	5.0	0.078	5.0	5.00
Ent.ment	gmiGA-wrap	0.600	1.0	3.151	2.0	0.055	2.0	0.539	2.0	0.118	3.0	2.00
	gmi-ML-CFS	0.593	3.0	3.158	3.0	0.056	3.0	0.548	3.0	0.119	4.0	3.20
	NoFS	0.597	2.0	3.135	1.0	0.056	4.0	0.537	1.0	0.116	1.0	1.80
	BR(kNN)	0.465	6.0	3.984	6.0	0.281	6.0	0.715	6.0	0.159	6.0	6.00
	CFS-U	0.583	4.0	3.194	4.0	0.055	1.0	0.548	4.0	0.118	2.0	3.00
	RFML	0.486	5.0	3.925	5.0	0.065	5.0	0.690	5.0	0.155	5.0	5.00
Computer	gmiGA-wrap	0.625	3.0	4.393	3.0	0.040	1.0	0.442	1.0	0.093	3.0	2.20
	gmi-ML-CFS	0.623	4.0	4.416	4.0	0.040	2.0	0.450	4.0	0.094	4.0	3.60
	NoFS	0.630	2.0	4.289	1.0	0.040	3.5	0.443	3.0	0.091	2.0	2.30
	BR(kNN)	0.599	6.0	4.840	6.0	0.112	6.0	0.475	6.0	0.101	6.0	6.00
	CFS-U	0.631	1.0	4.291	2.0	0.040	3.5	0.442	2.0	0.091	1.0	1.90
	RFML	0.606	5.0	4.591	5.0	0.043	5.0	0.474	5.0	0.099	5.0	5.00
Science	gmiGA-wrap	0.458	3.0	6.937	3.0	0.035	4.0	0.672	3.0	0.136	3.0	3.20
	gmi-ML-CFS	0.463	1.0	6.965	4.0	0.034	1.0	0.662	1.0	0.137	4.0	2.20
	NoFS	0.456	4.0	6.852	2.0	0.035	3.0	0.676	4.0	0.134	2.0	3.00
	BR(kNN)	0.391	6.0	8.112	6.0	0.236	6.0	0.758	6.0	0.165	6.0	6.00
	CFS-U	0.462	2.0	6.812	1.0	0.035	2.0	0.668	2.0	0.133	1.0	1.60
	RFML	0.415	5.0	7.258	5.0	0.036	5.0	0.729	5.0	0.144	5.0	5.00
MEAN	gmiGA-wrap		2.6		2.9		2.4		2.7		2.7	2.66
	gmi-ML-CFS		3.1		3.3		2.6		2.9		3.8	3.12
	NoFS		2.9		2.3		3.2		3.2		2.4	2.79
	BR(kNN)		5.5		5.4		5.8		5.6		5.4	5.54
	CFS-U		2.2		2.3		2.5		2.4		2.0	2.25
	RFML		4.7		4.8		4.6		4.3		4.8	4.64

Table 6.23: Values of five multi-label predictive accuracy measures for ML-kNN classifier with six different multi-label feature selection methods - feature space size = 200

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiGA-wrap	0.585	3.0	13.570	5.0	0.058	2.0	0.397	4.0	0.098	4.0	3.60
	gmi-ML-CFS	0.559	6.0	13.293	1.0	0.087	5.0	0.405	5.0	0.098	3.0	4.00
	NoFS	0.596	1.0	13.404	3.5	0.057	1.0	0.373	1.0	0.097	2.0	1.70
	BR(kNN)	0.566	5.0	14.288	6.0	0.094	6.0	0.413	6.0	0.103	6.0	5.80
	CFS-U	0.589	2.0	13.325	2.0	0.058	3.0	0.383	2.0	0.096	1.0	2.00
	RFML	0.582	4.0	13.404	3.5	0.059	4.0	0.394	3.0	0.101	5.0	3.90
Medical	gmiGA-wrap	0.797	4.0	3.017	4.0	0.016	3.0	0.254	4.0	0.048	4.0	3.80
	gmi-ML-CFS	0.820	2.0	2.772	2.0	0.015	1.0	0.225	1.0	0.045	2.0	1.60
	NoFS	0.745	6.0	3.557	6.0	0.019	6.0	0.321	6.0	0.060	6.0	6.00
	BR(kNN)	0.825	1.0	2.228	1.0	0.016	2.0	0.231	2.0	0.033	1.0	1.40
	CFS-U	0.769	5.0	3.242	5.0	0.018	5.0	0.292	5.0	0.053	5.0	5.00
	RFML	0.802	3.0	2.890	3.0	0.017	4.0	0.251	3.0	0.045	3.0	3.20
Business	gmiGA-wrap	0.873	3.0	2.331	3.0	0.028	4.0	0.126	4.0	0.042	4.0	3.60
	gmi-ML-CFS	0.853	5.0	2.751	6.0	0.041	5.0	0.139	5.0	0.049	6.0	5.40
	NoFS	0.876	2.0	2.299	2.0	0.028	1.5	0.124	2.0	0.041	2.0	1.90
	BR(kNN)	0.853	6.0	2.726	5.0	0.042	6.0	0.139	6.0	0.049	5.0	5.60
	CFS-U	0.877	1.0	2.262	1.0	0.028	1.5	0.124	1.0	0.040	1.0	1.10
	RFML	0.873	4.0	2.352	4.0	0.028	3.0	0.126	3.0	0.042	3.0	3.40
Art	gmiGA-wrap	0.536	2.0	5.324	3.0	0.059	2.0	0.578	2.0	0.147	3.0	2.40
	gmi-ML-CFS	0.437	5.0	5.985	4.0	0.185	5.0	0.752	5.5	0.175	4.0	4.70
	NoFS	0.519	3.0	5.319	2.0	0.059	3.0	0.605	3.0	0.147	2.0	2.60
	BR(kNN)	0.414	6.0	7.523	6.0	0.557	6.0	0.752	5.5	0.226	6.0	5.90
	CFS-U	0.541	1.0	5.190	1.0	0.058	1.0	0.572	1.0	0.141	1.0	1.00
	RFML	0.442	4.0	6.081	5.0	0.064	4.0	0.735	4.0	0.176	5.0	4.40
Education	gmiGA-wrap	0.551	1.0	3.896	3.0	0.041	2.0	0.588	1.0	0.092	3.0	2.00
	gmi-ML-CFS	0.480	5.0	4.477	5.0	0.142	5.0	0.681	5.5	0.107	5.0	5.10
	NoFS	0.544	3.0	3.895	2.0	0.041	3.0	0.602	3.0	0.092	2.0	2.60
	BR(kNN)	0.467	6.0	5.435	6.0	0.271	6.0	0.681	5.5	0.125	6.0	5.90
	CFS-U	0.549	2.0	3.876	1.0	0.041	1.0	0.592	2.0	0.091	1.0	1.40
	RFML	0.489	4.0	4.352	4.0	0.044	4.0	0.673	4.0	0.105	4.0	4.00
Recreation	gmiGA-wrap	0.572	1.0	4.171	2.0	0.055	1.0	0.545	2.0	0.151	1.0	1.40
	gmi-ML-CFS	0.379	5.0	5.530	5.0	0.206	5.0	0.803	5.0	0.219	5.0	5.00
	NoFS	0.553	3.0	4.321	3.0	0.056	3.0	0.570	3.0	0.158	3.0	3.00
	BR(kNN)	0.314	6.0	7.562	6.0	0.559	6.0	0.803	6.0	0.311	6.0	6.00
	CFS-U	0.571	2.0	4.166	1.0	0.055	2.0	0.540	1.0	0.152	2.0	1.60
	RFML	0.426	4.0	4.931	4.0	0.064	4.0	0.745	4.0	0.189	4.0	4.00
Health	gmiGA-wrap	0.687	1.0	3.411	2.0	0.042	1.0	0.387	1.0	0.064	2.0	1.40
	gmi-ML-CFS	0.617	5.0	3.976	5.0	0.113	5.0	0.489	5.5	0.077	5.0	5.10
	NoFS	0.673	3.0	3.453	3.0	0.044	3.0	0.412	4.0	0.065	3.0	3.20
	BR(kNN)	0.607	6.0	4.037	6.0	0.158	6.0	0.489	5.5	0.081	6.0	5.90
	CFS-U	0.684	2.0	3.380	1.0	0.043	2.0	0.402	2.0	0.063	1.0	1.60
	RFML	0.669	4.0	3.573	4.0	0.045	4.0	0.412	3.0	0.067	4.0	3.80
Ent.ment	gmiGA-wrap	0.618	2.0	3.056	3.0	0.054	1.0	0.498	1.0	0.111	2.0	1.80
	gmi-ML-CFS	0.506	4.0	3.533	4.0	0.172	5.0	0.688	5.0	0.135	4.0	4.40
	NoFS	0.624	1.0	2.982	1.0	0.056	3.0	0.500	2.0	0.108	1.0	1.60
	BR(kNN)	0.451	6.0	4.843	6.0	0.460	6.0	0.715	6.0	0.192	6.0	6.00
	CFS-U	0.613	3.0	3.049	2.0	0.054	2.0	0.513	3.0	0.111	3.0	2.60
	RFML	0.489	5.0	3.887	5.0	0.065	4.0	0.688	4.0	0.152	5.0	4.60
Computer	gmiGA-wrap	0.643	3.0	4.178	3.0	0.038	3.0	0.430	3.0	0.089	3.0	3.00
	gmi-ML-CFS	0.601	5.0	4.810	5.0	0.084	5.0	0.475	5.5	0.101	5.0	5.10
	NoFS	0.647	2.0	4.125	2.0	0.038	2.0	0.424	2.0	0.087	1.5	1.90
	BR(kNN)	0.589	6.0	5.099	6.0	0.160	6.0	0.475	5.5	0.110	6.0	5.90
	CFS-U	0.648	1.0	4.115	1.0	0.038	1.0	0.423	1.0	0.087	1.5	1.10
	RFML	0.609	4.0	4.456	4.0	0.042	4.0	0.474	4.0	0.095	4.0	4.00
Science	gmiGA-wrap	0.482	2.0	6.792	3.0	0.034	2.0	0.638	1.0	0.133	3.0	2.20
	gmi-ML-CFS	0.396	5.0	7.811	5.0	0.129	5.0	0.758	5.5	0.157	5.0	5.10
	NoFS	0.476	3.0	6.617	2.0	0.034	3.0	0.654	3.0	0.129	2.0	2.60
	BR(kNN)	0.386	6.0	8.877	6.0	0.490	6.0	0.758	5.5	0.182	6.0	5.90
	CFS-U	0.487	1.0	6.563	1.0	0.034	1.0	0.640	2.0	0.127	1.0	1.20
	RFML	0.441	4.0	7.070	4.0	0.036	4.0	0.690	4.0	0.140	4.0	4.00
MEAN	gmiGA-wrap		2.2		3.1		2.1		2.3		2.9	2.52
	gmi-ML-CFS		4.7		4.2		4.6		4.9		4.4	4.55
	NoFS		2.7		2.7		2.9		2.9		2.5	2.71
	BR(kNN)		5.4		5.4		5.6		5.4		5.4	5.43
	CFS-U		2.0		1.6		2.0		2.0		1.8	1.86
	RFML		4.0		4.1		3.9		3.6		4.1	3.93

Table 6.24: Values of five multi-label predictive accuracy measures for ML-kNN classifier with six different multi-label feature selection methods - feature space size = 300

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiGA-wrap	0.590	1.0	13.391	1.0	0.05702	1.0	0.382	1.0	0.098	1.0	1.00
	gmi-ML-CFS	0.581	2.0	13.432	2.0	0.058	2.0	0.406	5.0	0.098	2.0	2.60
	NoFS	0.567	4.0	13.629	5.0	0.059	5.0	0.404	4.0	0.100	5.0	4.60
	BR(kNN)	0.554	6.0	14.808	6.0	0.147	6.0	0.508	6.0	0.113	6.0	6.00
	CFS-U	0.567	5.0	13.584	4.0	0.058	3.0	0.396	2.0	0.100	3.0	3.40
	RFML	0.577	3.0	13.560	3.0	0.059	4.0	0.397	3.0	0.100	4.0	3.40
Medical	gmiGA-wrap	0.805	3.0	2.899	4.0	0.01622	1.0	0.245	3.0	0.046	3.0	2.80
	gmi-ML-CFS	0.819	1.0	2.831	3.0	0.016	2.0	0.225	1.0	0.044	2.0	1.80
	NoFS	0.738	5.0	3.578	6.0	0.019	5.0	0.336	5.0	0.060	6.0	5.40
	BR(kNN)	0.694	6.0	2.816	2.0	0.028	6.0	0.411	6.0	0.047	4.0	4.80
	CFS-U	0.776	4.0	3.222	5.0	0.018	4.0	0.292	4.0	0.052	5.0	4.40
	RFML	0.815	2.0	2.766	1.0	0.017	3.0	0.233	2.0	0.043	1.0	1.80
Business	gmiGA-wrap	0.875	4.0	2.294	4.0	0.0283	4.0	0.126	3.0	0.041	3.0	3.60
	gmi-ML-CFS	0.876	3.0	2.292	3.0	0.028	3.0	0.127	4.0	0.040	1.5	2.90
	NoFS	0.876	2.0	2.288	2.0	0.028	1.0	0.124	2.0	0.041	4.0	2.20
	BR(kNN)	0.854	6.0	2.736	6.0	0.044	6.0	0.139	6.0	0.049	6.0	6.00
	CFS-U	0.877	1.0	2.280	1.0	0.028	2.0	0.123	1.0	0.040	1.5	1.30
	RFML	0.871	5.0	2.339	5.0	0.029	5.0	0.131	5.0	0.043	5.0	5.00
Art	gmiGA-wrap	0.533	3.0	5.325	4.0	0.0585	2.0	0.584	3.0	0.146	4.0	3.20
	gmi-ML-CFS	0.540	2.0	5.278	3.0	0.058	1.0	0.575	2.0	0.145	3.0	2.20
	NoFS	0.521	4.0	5.256	2.0	0.060	4.0	0.607	4.0	0.144	2.0	3.20
	BR(kNN)	0.234	6.0	8.567	6.0	0.627	6.0	0.978	6.0	0.269	6.0	6.00
	CFS-U	0.543	1.0	5.108	1.0	0.059	3.0	0.572	1.0	0.139	1.0	1.40
	RFML	0.440	5.0	6.068	5.0	0.064	5.0	0.738	5.0	0.175	5.0	5.00
Education	gmiGA-wrap	0.558	1.0	3.817	1.0	0.04056	1.0	0.582	1.0	0.089	1.0	1.00
	gmi-ML-CFS	0.552	2.0	3.895	3.0	0.041	2.0	0.588	2.0	0.091	3.0	2.40
	NoFS	0.541	4.0	3.914	4.0	0.041	4.0	0.604	4.0	0.092	4.0	4.00
	BR(kNN)	0.151	6.0	10.153	6.0	0.470	6.0	0.987	6.0	0.284	6.0	6.00
	CFS-U	0.548	3.0	3.877	2.0	0.041	3.0	0.596	3.0	0.091	2.0	2.60
	RFML	0.488	5.0	4.359	5.0	0.044	5.0	0.672	5.0	0.105	5.0	5.00
Recreation	gmiGA-wrap	0.586	1.0	4.072	1.0	0.0546	2.0	0.527	1.0	0.147	2.0	1.40
	gmi-ML-CFS	0.581	2.0	4.147	3.0	0.054	1.0	0.530	2.0	0.150	3.0	2.20
	NoFS	0.552	4.0	4.296	4.0	0.056	4.0	0.573	4.0	0.157	4.0	4.00
	BR(kNN)	0.154	6.0	9.750	6.0	0.674	6.0	0.995	6.0	0.414	6.0	6.00
	CFS-U	0.576	3.0	4.074	2.0	0.055	3.0	0.542	3.0	0.147	1.0	2.40
	RFML	0.446	5.0	4.708	5.0	0.063	5.0	0.722	5.0	0.179	5.0	5.00
Health	gmiGA-wrap	0.687	2.0	3.360	2.0	0.04256	2.0	0.397	2.0	0.063	2.0	2.00
	gmi-ML-CFS	0.699	1.0	3.303	1.0	0.042	1.0	0.380	1.0	0.061	1.0	1.00
	NoFS	0.674	4.0	3.441	4.0	0.045	4.0	0.418	4.0	0.065	4.0	4.00
	BR(kNN)	0.602	6.0	4.386	6.0	0.220	6.0	0.489	6.0	0.089	6.0	6.00
	CFS-U	0.682	3.0	3.373	3.0	0.044	3.0	0.407	3.0	0.063	3.0	3.00
	RFML	0.662	5.0	3.581	5.0	0.045	5.0	0.429	5.0	0.068	5.0	5.00
Ent.ment	gmiGA-wrap	0.628	1.0	2.971	1.0	0.05378	1.0	0.490	1.0	0.108	1.0	1.00
	gmi-ML-CFS	0.609	3.0	3.023	3.0	0.054	2.0	0.529	4.0	0.111	4.0	3.20
	NoFS	0.608	4.0	3.034	4.0	0.057	4.0	0.523	3.0	0.111	3.0	3.60
	BR(kNN)	0.211	6.0	7.262	6.0	0.513	6.0	0.923	6.0	0.324	6.0	6.00
	CFS-U	0.612	2.0	2.975	2.0	0.055	3.0	0.517	2.0	0.108	2.0	2.20
	RFML	0.493	5.0	3.807	5.0	0.064	5.0	0.685	5.0	0.148	5.0	5.00
Computer	gmiGA-wrap	0.647	3.0	4.164	4.0	0.0375	4.0	0.426	3.0	0.088	4.0	3.60
	gmi-ML-CFS	0.646	4.0	4.161	3.0	0.038	3.0	0.427	4.0	0.088	3.0	3.40
	NoFS	0.651	1.0	4.086	2.0	0.037	2.0	0.423	1.0	0.086	2.0	1.60
	BR(kNN)	0.251	6.0	8.628	6.0	0.507	6.0	0.939	6.0	0.205	6.0	6.00
	CFS-U	0.651	2.0	4.067	1.0	0.037	1.0	0.424	2.0	0.086	1.0	1.40
	RFML	0.619	5.0	4.439	5.0	0.041	5.0	0.456	5.0	0.094	5.0	5.00
Science	gmiGA-wrap	0.481	2.0	6.628	4.0	0.03388	1.0	0.645	2.0	0.129	3.0	2.40
	gmi-ML-CFS	0.489	1.0	6.622	3.0	0.034	2.0	0.629	1.0	0.129	2.0	1.80
	NoFS	0.475	4.0	6.611	2.0	0.034	3.0	0.660	4.0	0.130	4.0	3.40
	BR(kNN)	0.119	6.0	14.552	6.0	0.559	6.0	0.967	6.0	0.332	6.0	6.00
	CFS-U	0.477	3.0	6.535	1.0	0.035	4.0	0.657	3.0	0.128	1.0	2.40
	RFML	0.430	5.0	7.047	5.0	0.036	5.0	0.711	5.0	0.140	5.0	5.00
MEAN	gmiGA-wrap		2.1		2.6		1.9		2.0		2.4	2.20
	gmi-ML-CFS		2.1		2.7		1.9		2.6		2.5	2.35
	NoFS		3.6		3.5		3.6		3.5		3.8	3.60
	BR(kNN)		6.0		5.6		6.0		6.0		5.8	5.88
	CFS-U		2.7		2.2		2.9		2.4		2.1	2.45
	RFML		4.5		4.4		4.7		4.5		4.5	4.52

Table 6.25: Values of five multi-label predictive accuracy measures for ML-kNN classifier with six different multi-label feature selection methods - feature space size = 400

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiGA-wrap	0.580	5.0	13.358	1.0	0.05762	5.0	0.406	5.0	0.098	3.0	3.80
	gmi-ML-CFS	0.587	2.0	13.359	2.0	0.057	4.0	0.404	4.0	0.097	2.0	2.80
	NoFS	0.583	3.0	13.40	4.0	0.056	1.0	0.382	2.0	0.098	4.0	2.80
	BR(kNN)	0.471	6.0	14.22	6.0	0.165	6.0	0.760	6.0	0.113	6.0	6.00
	CFS-U	0.580	4.0	13.47	5.0	0.057	2.0	0.385	3.0	0.099	5.0	3.80
	RFML	0.608	1.0	13.383	3.0	0.057	3.0	0.378	1.0	0.096	1.0	1.80
Medical	gmiGA-wrap	0.796	2.0	3.030	1.0	0.01694	3.0	0.258	2.0	0.049	1.0	1.80
	gmi-ML-CFS	0.785	3.0	3.186	3.0	0.017	2.0	0.271	3.0	0.052	3.0	2.80
	NoFS	0.728	5.0	3.72	5.0	0.020	5.0	0.349	5.0	0.063	5.0	5.00
	BR(kNN)	0.110	6.0	13.81	6.0	0.420	6.0	0.980	6.0	0.291	6.0	6.00
	CFS-U	0.768	4.0	3.34	4.0	0.019	4.0	0.295	4.0	0.055	4.0	4.00
	RFML	0.801	1.0	3.181	2.0	0.016	1.0	0.247	1.0	0.051	2.0	1.40
Business	gmiGA-wrap	0.876	4.0	2.288	4.0	0.02826	4.0	0.125	4.0	0.041	4.0	4.00
	gmi-ML-CFS	0.876	3.0	2.286	3.0	0.028	3.0	0.125	3.0	0.040	3.0	3.00
	NoFS	0.881	1.0	2.26	2.0	0.028	1.0	0.119	1.0	0.039	1.0	1.20
	BR(kNN)	0.767	6.0	4.01	6.0	0.294	6.0	0.139	6.0	0.075	6.0	6.00
	CFS-U	0.879	2.0	2.24	1.0	0.028	2.0	0.123	2.0	0.039	2.0	1.80
	RFML	0.871	5.0	2.332	5.0	0.028	5.0	0.132	5.0	0.042	5.0	5.00
Art	gmiGA-wrap	0.532	3.0	5.296	3.0	0.058	2.0	0.587	3.0	0.145	3.0	2.80
	gmi-ML-CFS	0.535	2.0	5.251	2.0	0.058	3.0	0.585	2.0	0.144	2.0	2.20
	NoFS	0.509	4.0	5.34	4.0	0.060	4.0	0.631	4.0	0.147	4.0	4.00
	BR(kNN)	0.150	6.0	12.52	6.0	0.468	6.0	0.980	6.0	0.424	6.0	6.00
	CFS-U	0.546	1.0	5.08	1.0	0.058	1.0	0.569	1.0	0.137	1.0	1.00
	RFML	0.451	5.0	5.981	5.0	0.064	5.0	0.715	5.0	0.171	5.0	5.00
Education	gmiGA-wrap	0.555	2.0	3.818	2.0	0.04068	1.0	0.585	2.0	0.089	2.0	1.80
	gmi-ML-CFS	0.555	1.0	3.826	3.0	0.041	2.5	0.583	1.0	0.090	3.0	2.10
	NoFS	0.535	4.0	3.95	4.0	0.042	4.0	0.611	4.0	0.093	4.0	4.00
	BR(kNN)	0.143	6.0	9.95	6.0	0.508	6.0	0.999	6.0	0.272	6.0	6.00
	CFS-U	0.555	3.0	3.78	1.0	0.041	2.5	0.589	3.0	0.089	1.0	2.10
	RFML	0.494	5.0	4.305	5.0	0.044	5.0	0.665	5.0	0.103	5.0	5.00
Recreation	gmiGA-wrap	0.583	1.0	4.067	2.0	0.0546	1.0	0.533	1.0	0.147	2.0	1.40
	gmi-ML-CFS	0.378	5.0	5.703	5.0	0.065	5.0	0.805	5.0	0.220	5.0	5.00
	NoFS	0.552	3.0	4.24	3.0	0.057	3.0	0.576	3.0	0.155	3.0	3.00
	BR(kNN)	0.176	6.0	10.93	6.0	0.684	6.0	0.949	6.0	0.452	6.0	6.00
	CFS-U	0.578	2.0	4.06	1.0	0.055	2.0	0.539	2.0	0.147	1.0	1.60
	RFML	0.453	4.0	4.598	4.0	0.063	4.0	0.712	4.0	0.174	4.0	4.00
Health	gmiGA-wrap	0.714	1.0	3.204	1.0	0.04056	1.0	0.356	1.0	0.058	1.0	1.00
	gmi-ML-CFS	0.708	2.0	3.229	2.0	0.041	2.0	0.366	2.0	0.059	2.0	2.00
	NoFS	0.692	4.0	3.30	4.0	0.043	4.0	0.395	4.0	0.061	4.0	4.00
	BR(kNN)	0.378	6.0	5.17	6.0	0.303	6.0	0.957	6.0	0.114	6.0	6.00
	CFS-U	0.701	3.0	3.25	3.0	0.043	3.0	0.378	3.0	0.060	3.0	3.00
	RFML	0.676	5.0	3.472	5.0	0.045	5.0	0.414	5.0	0.065	5.0	5.00
Ent.ment	gmiGA-wrap	0.636	1.0	2.915	2.0	0.0539	1.0	0.484	1.0	0.105	2.0	1.40
	gmi-ML-CFS	0.631	2.0	2.936	3.0	0.054	2.0	0.491	2.0	0.106	3.0	2.40
	NoFS	0.617	4.0	3.00	4.0	0.057	4.0	0.510	4.0	0.110	4.0	4.00
	BR(kNN)	0.221	6.0	6.82	6.0	0.567	6.0	0.961	6.0	0.297	6.0	6.00
	CFS-U	0.630	3.0	2.89	1.0	0.054	3.0	0.495	3.0	0.105	1.0	2.20
	RFML	0.511	5.0	3.645	5.0	0.064	5.0	0.657	5.0	0.139	5.0	5.00
Computer	gmiGA-wrap	0.647	4.0	4.108	3.0	0.03738	3.0	0.431	4.0	0.087	3.0	3.40
	gmi-ML-CFS	0.648	3.0	4.137	4.0	0.037	4.0	0.426	3.0	0.087	4.0	3.60
	NoFS	0.655	2.0	4.03	2.0	0.037	2.0	0.418	2.0	0.084	2.0	2.00
	BR(kNN)	0.213	6.0	8.45	6.0	0.584	6.0	0.967	6.0	0.213	6.0	6.00
	CFS-U	0.655	1.0	4.01	1.0	0.037	1.0	0.417	1.0	0.084	1.0	1.00
	RFML	0.628	5.0	4.307	5.0	0.040	5.0	0.448	5.0	0.091	5.0	5.00
Science	gmiGA-wrap	0.484	1.0	6.716	3.0	0.03402	1.5	0.635	1.0	0.130	2.0	1.70
	gmi-ML-CFS	0.479	3.0	6.717	4.0	0.034	1.5	0.641	2.0	0.131	3.0	2.70
	NoFS	0.462	4.0	6.68	2.0	0.035	4.0	0.671	4.0	0.132	4.0	3.60
	BR(kNN)	0.145	6.0	13.28	6.0	0.593	6.0	0.980	6.0	0.293	6.0	6.00
	CFS-U	0.482	2.0	6.53	1.0	0.034	3.0	0.648	3.0	0.128	1.0	2.00
	RFML	0.438	5.0	7.006	5.0	0.036	5.0	0.697	5.0	0.139	5.0	5.00
MEAN	gmiGA-wrap		2.4		2.2		2.3		2.4		2.3	2.31
	gmi-ML-CFS		2.6		3.1		2.9		2.7		3.0	2.86
	NoFS		3.4		3.4		3.2		3.3		3.5	3.36
	BR(kNN)		6.0		6.0		6.0		6.0		6.0	6.00
	CFS-U		2.5		1.9		2.4		2.5		2.0	2.25
	RFML		4.1		4.4		4.3		4.1		4.2	4.22

Table 6.26: Summary of results in terms of average ranking (AR) and the number of selected features (Sel.F) of gmiGA-wrap and other multi-label feature selection methods (using ML-kNN as the classifier)

FS.size	**NoFS**		**BR(kNN)**		**CFS-U**		**RFML**		**gmi-ML-CFS**		**gmiGA-wrap**	
	AR	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**
100	2.79	100.00	5.54	100.00	**2.25**	73.90	4.64	25.60	3.12	22.40	2.66	25.60
200	2.71	200.00	5.43	200.00	**1.86**	128.40	3.93	52.26	4.55	34.30	2.52	52.26
300	3.60	300.00	5.88	300.00	2.45	174.80	4.52	83.60	2.35	44.10	**2.20**	83.60
400	3.36	400.00	6.00	400.00	**2.25**	214.40	4.22	118.86	2.86	57.00	2.31	118.86
Avg	3.12	250.00	5.71	250.00	**2.20**	147.88	4.33	70.08	3.22	39.45	2.42	70.08

the corresponding feature space size – i.e, Table 6.22 for feature space size 100, etc.

Regarding predictive accuracy, CFS-U obtains the best average rank (2.20) across all feature space sizes (last row of Table 6.26); and it is the winner in 3 out of 4 feature space sizes. The only exception is feature space size 300, where gmiGA-wrap was the winner.

The difference between the average ranks of CFS-U and gmiML-CFS was small in most cases (between 0.06 – 0.41) except when feature space sizes equals to 200: in this case the difference is 0.66 (2.52 – 1.86) as shown in Table 6.26.

However, CFS-U has the disadvantage of selecting a much larger number of features than the other three feature selection method (RFML, gmiML-CFS and gmiGA-wrap). For example; when the individual length is equal to 400, CFS-U obtain the largest selected feature subset (214.4 features), which is almost twice the number of features selected by gmiGA-wrap (118.86 features).

Figure 6.3 shows the overall average ranking (AR) for gmiGA-wrap and the other multi-label feature selection methods plotted against the average number of selected features across all datasets and feature space sizes, when using ML-kNN as the classifier. Clearly, gmiGA-wrap obtained a very good trade-off between minimizing average ranking and minimizing the number of selected features. In

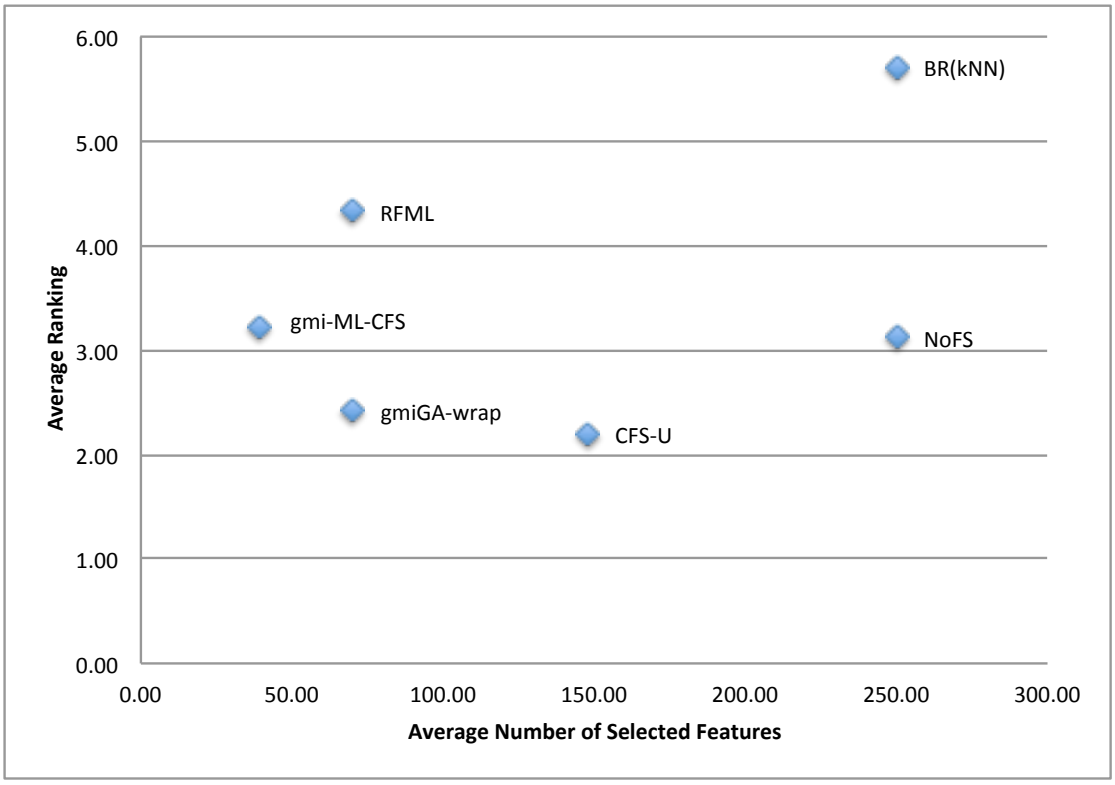


Figure 6.3: Overall average ranking (AR) for gmiGA-wrap and the other multi-label feature selection methods plotted against the average number of selected features across all datasets and feature space sizes, when using ML-kNN as the classifier

particular, gmiGA-wrap was only slightly worse than CFS-U in term of average ranking, but gmiGA-wrap was substantially better than CFS-U in terms of the number of selected features.

Moreover, RFML, which has the same size of selected feature subset as gmiGA-wrap, obtains much worse average rank than gmiGA-wrap; while NoFS and BR, which use the full set of input features, still obtain a larger average rank than gmiGA-wrap.

In general, gmiGA-wrap obtained the second best average rank among the six multi-label feature selection approaches compared in Table 6.26. gmiGA-wrap was outperformed only by CFS-U, which obtained an average rank of 2.20, slightly

Table 6.27: Summary of overall average ranking (AR) across four individual lengths for gmiGA-wrap and other Multi-Label feature Selection methods using ML-kNN as classifier

Dataset	**Overall Average Rank (AR) across 4 individual lengths**					
	gmiGA-wrap	**gmi-ML-CFS**	**NoFS**	**BR(kNN)**	**CFS-U**	**RFML**
Enron	3.00(3.5)	3.00(3.5)	2.93(2)	5.95(6)	2.80(1)	3.33(5)
Medical	2.50(2)	2.53(3)	5.60(6)	3.50(4)	4.58(5)	2.30(1)
Business	3.55(4)	3.50(3)	1.98(2)	5.90(6)	1.48(1)	4.60(5)
Art	2.95(3)	2.88(2)	3.15(4)	5.88(6)	1.20(1)	4.95(5)
Education	1.70(1)	3.65(4)	3.33(3)	5.98(6)	1.85(2)	4.50(5)
Recreation	1.55(1)	3.90(4)	2.95(3)	6.00(6)	2.10(2)	4.50(5)
Health	2.00(1)	2.58(3)	3.38(4)	5.98(6)	2.38(2)	4.70(5)
Ent.ment	1.55(1)	3.30(4)	2.75(3)	6.00(6)	2.50(2)	4.90(5)
Computer	3.05(3)	3.93(4)	1.95(2)	5.98(6)	1.35(1)	4.75(5)
Science	2.38(2)	2.95(3)	3.15(4)	5.98(6)	1.80(1)	4.75(5)
Average	2.42(2.15)	3.22(3.35)	3.12(3.3)	5.71(5.8)	**2.2(1.8)**	4.33(4.6)

smaller than gmiGA-wrap's average rank (2.42). That is, gmiGA-wrap obtained substantially better predictive accuracy (substantially lower overall average rank across all datasets and all accuracy measures) than gmiML-CFS, NoFS, RFML and BR.

Table 6.27 presents a summary of the results from another perspective, reporting the average ranks (in terms of predictive accuracy) for each dataset, averaged across the 4 GA individual lengths (feature space sizes). In each cell of the table, the first value is the average rank computed by averaging the corresponding ranks in Tables 6.22 - 6.25; whilst the value between brackets is the "rank of the average ranks". This latter value was use for the statistical tests of significance.

Using the results shows in Table 6.27, we run the Friedman test and confidently conclude that there is a significant difference among the 6 methods on the 10 evaluation datasets at the 0.05 level of significance for a two tailed test (p value is 0.00001). Running the Holm's posthoc test on these results using gmiGA-wrap as the control method, there are no significant differences when comparing gmiGA-wrap versus CFS-U, NoFS, and gmiML-CFS at the 0.05 significance level, but there is a significant difference between gmiGA-wrap and BR, as well as between

gmiGA-wrap and and RFML at the same level of significance (p value = 0.00018 and 0.03749, respectively).

Table 6.28 shows GA-ML-CFS' running time on three multi-label evaluation datasets: Enron, Entertainment and Health. These datasets were selected based on their number of labels. The Enron dataset has the largest number of labels (53 labels) among the 10 evaluation datasets, the Entertainment dataset has the smallest number of labels (21 labels) and the Health dataset has an intermediate number of labels (32 labels). For each of those datasets, the table reports the running time for 4 versions of the datasets, varying the number of input features from 100 to 400.

In this Table, the second column shows the running time of GA-ML-CFS, the third column shows the time for running ML-kNN using only the features selected by GA-ML-CFS, the fourth column shows the summation of the previous two running times, the fifth column shows the running time of CFS-U, the sixth column shows the time for running ML-kNN using the features selected by CFS-U and the last column shows the summation of the previous two running times (CFS-U and ML-kNN). The running time format (d:h:m:s) shown in Table 6.28 refers to days, hours, minutes and seconds. Note that all experiments which measure the computational time were run on a system with Intel Core i7 CPU at 3.40 GHz and 16.0 GB of memory.

Clearly, the longest time for running GA-ML-CFS was obtained in the Entertainment dataset. The number of instances of the Entertainment dataset is relatively large (12,730 instances). In general, the ML-kNN-running time is very small when using the features selected by GA-ML-CFS. The longest ML-kNN-running time is 58 seconds on the biggest dataset (Entertainment 400). In our experiments the running time of the CFS-U approach is shorter than the GA-running time. However, if we compare the running time of ML-kNN using different sets of

Table 6.28: Comparing the computational time of GA-ML-CFS and CFS-U with ML-kNN on three different datasets

Dataset	Running Time (d:h:m:s)					
	GA-ML-CFS	ML-kNN with features selected by GA-ML-CFS	ML-kNN and GA-ML-CFS	CFS-U	ML-kNN with features selected by CFS-U	CFS-U and ML-kNN
Enron 100	0:01:15:12	0:00:00:01	0:01:15:13	0:01:01:10	0:00:00:02	0:01:01:12
Enron 200	0:03:21:41	0:00:00:01	0:03:21:42	0:01:02:36	0:00:00:03	0:01:02:39
Enron 300	0:06:23:48	0:00:00:02	0:06:23:50	0:01:09:23	0:00:00:03	0:01:09:26
Enron 400	0:12:23:50	0:00:00:03	0:12:23:53	0:01:23:01	0:00:00:03	0:01:23:04
Entertainment 100	0:08:48:58	0:00:00:27	0:08:49:25	0:01:01:07	0:00:00:33	0:01:01:40
Entertainment 200	1:08:11:34	0:00:00:39	1:08:12:13	0:01:03:33	0:00:00:47	0:01:04:20
Entertainment 300	2:17:43:00	0:00:00:46	2:17:43:46	0:01:08:44	0:00:01:04	0:01:08:48
Entertainment 400	5:06:19:08	0:00:00:58	5:06:20:06	0:01:18:38	0:00:01:15	0:01:18:53
Health 100	0:02:44:12	0:00:00:15	0:02:44:27	0:01:01:34	0:00:00:39	0:01:02:13
Health 200	0:08:24:00	0:00:00:15	0:08:24:15	0:01:03:34	0:00:00:55	0:01:04:29
Health 300	0:20:30:13	0:00:00:12	0:20:30:25	0:01:09:11	0:00:01:04	0:01:09:15
Health 400	1:10:59:35	0:00:00:20	1:10:59:55	0:01:20:33	0:00:01:23	0:01:20:55

selected features, ML-kNN using the features selected by CFS-U took more time than ML-kNN using features selected by GA-ML-CFS. This because CFS-U selects many more features than GA-ML-CFS. On the other hand, the total time to run both CFS-U and ML-kNN is in general shorter than the total time to run both GA-ML-CFS and ML-kNN, since GA-ML-CFS is substantially more time consuming than CFS-U.

6.8.3 Results for the Best Version of GA-ML-CFS (gmiGA-wrap) and Other Multi-Label Feature Selection Methods using the BPMLL Classifier

The results are shown in Table 6.29 - 6.32, where the meaning of the columns are as explained in the beginning of Subsection 6.7.1. In Table 6.29, reporting results for the large datasets with feature space size equal to 100, BR obtained the best place with overall average rank 1.36; while gmiML-CFS obtained the second place with overall average rank 3.18. gmiGA-wrap obtained the third place with 3.27 average overall rank and outperformed CFS-U, RFML and NoFS with average rank 4.62, 4.70 and 3.87, respectively.

In Table 6.30, where the feature space size is equal to 200, again BR obtained the best result, with overall average rank 1.42. In addition, gmiGA-wrap was the second best method, with overall average rank = 2.86, and it outperformed gmiML-CFS, NoFS, CFS-U and RFML.

In Table 6.31, when the features space size is equal to 300, RFML was the best method with overall average rank = 3.22. In addition, gmiGA-wrap outperformed gmiML-CFS, CFS-U on all ten datasets, with overall average rank = 3.49.

In Table 6.32 again BR was the best method, with overall average rank =1.44 while gmiGA-wrap outperformed NoFS, CFS-U and RFML on all ten datasets, with overall average rank = 3.24 (RFML, CFS-U and NoFS obtained 3.12, 4.56 and 5.88 overall average rank respectively).

Table 6.33 reports, for each feature space size, the summary of results in terms of the overall average ranking and the number of selected features by the six approaches when using BPMLL as the classifier. BR(BPNN) obtains the best average rank (1.93), which is substantially better than the ranks of all other approaches. However, BR (like NoFS) uses all input features to train a computationally expensive BP neural net algorithm for each class label so, BR is a computationally expensive approach.

The second best method in Table 6.33 was gmiGA-wrap, with average rank 3.22. However, gmiGA-wrap selects on average 68.48 features, about 27.39% of the average of 250 features used by BR. So, the trainning of the BP neural net classifier with the features selected by gmiGA-wrap is substantially faster than the training of BPNN in the BR approach.

Figure 6.4 shows the overall average ranking (AR) for gmiGA-wrap and the

Table 6.29: Values of five multi-label predictive accuracy measures for BPMLL classifier with six different multi-label feature selection methods - feature space size = 100

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiGA-wrap	0.568	5.0	13.259	2.0	0.088	2.0	0.405	5.0	0.097	3.0	3.40
	gmi-ML-CFS	0.567	6.0	13.207	1.0	0.089	3.0	0.403	4.0	0.097	1.0	3.00
	NoFS	0.576	2.0	13.913	6.0	0.091	6.0	0.409	6.0	0.100	6.0	5.20
	BR(BPNN)	0.584	1.0	13.380	3.0	0.058	1.0	0.396	1.0	0.097	2.0	1.60
	CFS-U	0.573	3.0	13.811	5.0	0.090	5.0	0.397	2.0	0.100	5.0	4.00
	RFML	0.569	4.0	13.436	4.0	0.089	4.0	0.401	3.0	0.099	4.0	3.80
Medical	gmiGA-wrap	0.715	4.0	2.579	2.0	0.028	4.0	0.420	4.0	0.042	2.0	3.20
	gmi-ML-CFS	0.557	6.0	3.604	5.0	0.050	6.0	0.655	6.0	0.066	6.0	5.80
	NoFS	0.796	2.0	2.606	3.0	0.018	1.0	0.271	2.0	0.042	3.0	2.20
	BR(BPNN)	0.717	3.0	3.614	6.0	0.019	3.0	0.374	3.0	0.062	5.0	4.00
	CFS-U	0.805	1.0	2.296	1.0	0.018	2.0	0.265	1.0	0.035	1.0	1.20
	RFML	0.674	5.0	3.019	4.0	0.031	5.0	0.473	5.0	0.052	4.0	4.60
Business	gmiGA-wrap	0.852	4.0	2.768	4.0	0.043	3.0	0.139	4.5	0.049	4.0	3.90
	gmi-ML-CFS	0.853	3.0	2.751	3.0	0.042	2.0	0.139	2.0	0.048	2.0	2.40
	NoFS	0.853	2.0	2.730	2.0	0.043	4.0	0.139	4.5	0.049	3.0	3.10
	BR(BPNN)	0.874	1.0	2.369	1.0	0.028	1.0	0.124	1.0	0.043	1.0	1.00
	CFS-U	0.850	5.0	2.818	5.0	0.043	5.0	0.139	4.5	0.050	5.0	4.90
	RFML	0.849	6.0	2.857	6.0	0.044	6.0	0.139	4.5	0.050	6.0	5.70
Art	gmiGA-wrap	0.436	3.0	6.000	3.0	0.184	3.0	0.752	4.0	0.175	4.0	3.40
	gmi-ML-CFS	0.436	5.0	6.006	5.0	0.184	2.0	0.752	4.0	0.175	5.0	4.20
	NoFS	0.431	6.0	6.054	6.0	0.238	6.0	0.752	4.0	0.179	6.0	5.60
	BR(BPNN)	0.529	1.0	5.306	1.0	0.059	1.0	0.592	1.0	0.146	1.0	1.00
	CFS-U	0.438	2.0	5.909	2.0	0.218	5.0	0.752	4.0	0.172	2.0	3.00
	RFML	0.436	4.0	6.000	4.0	0.187	4.0	0.752	4.0	0.175	3.0	3.80
Education	gmiGA-wrap	0.479	3.0	4.560	3.0	0.120	2.0	0.680	3.0	0.108	3.0	2.80
	gmi-ML-CFS	0.480	2.0	4.532	2.0	0.134	5.0	0.679	2.0	0.107	2.0	2.60
	NoFS	0.476	5.0	4.697	5.0	0.146	6.0	0.681	5.0	0.111	5.0	5.20
	BR(BPNN)	0.543	1.0	3.938	1.0	0.041	1.0	0.602	1.0	0.093	1.0	1.00
	CFS-U	0.476	4.0	4.683	4.0	0.133	4.0	0.681	5.0	0.111	4.0	4.20
	RFML	0.475	6.0	4.710	6.0	0.128	3.0	0.681	5.0	0.112	6.0	5.20
Recreation	gmiGA-wrap	0.387	2.0	5.357	2.0	0.193	3.0	0.794	2.0	0.213	2.0	2.20
	gmi-ML-CFS	0.380	4.0	5.402	3.0	0.190	2.0	0.802	3.0	0.215	4.0	3.20
	NoFS	0.376	6.0	5.648	6.0	0.350	6.0	0.804	4.5	0.224	6.0	5.70
	BR(BPNN)	0.536	1.0	4.333	1.0	0.058	1.0	0.595	1.0	0.157	1.0	1.00
	CFS-U	0.381	3.0	5.447	4.0	0.224	5.0	0.804	4.5	0.215	3.0	3.90
	RFML	0.376	5.0	5.571	5.0	0.194	4.0	0.805	6.0	0.222	5.0	5.00
Health	gmiGA-wrap	0.621	3.0	3.935	3.0	0.111	3.0	0.485	3.0	0.077	3.0	3.00
	gmi-ML-CFS	0.623	2.0	3.927	2.0	0.108	2.0	0.481	2.0	0.076	2.0	2.00
	NoFS	0.612	5.0	4.040	6.0	0.130	4.0	0.489	4.0	0.079	6.0	5.00
	BR(BPNN)	0.631	1.0	3.784	1.0	0.049	1.0	0.476	1.0	0.075	1.0	1.00
	CFS-U	0.611	6.0	4.024	5.0	0.130	5.0	0.489	5.5	0.078	5.0	5.30
	RFML	0.617	4.0	3.999	4.0	0.130	6.0	0.489	5.5	0.078	4.0	4.70
Ent.ment	gmiGA-wrap	0.528	3.0	3.467	4.0	0.154	3.0	0.649	3.0	0.132	4.0	3.40
	gmi-ML-CFS	0.529	2.0	3.460	2.0	0.149	2.0	0.649	2.0	0.132	2.0	2.00
	NoFS	0.495	5.0	3.547	5.0	0.233	6.0	0.715	6.0	0.137	5.0	5.40
	BR(BPNN)	0.597	1.0	3.135	1.0	0.056	1.0	0.537	1.0	0.116	1.0	1.00
	CFS-U	0.523	4.0	3.460	3.0	0.162	4.0	0.662	4.0	0.132	3.0	3.60
	RFML	0.473	6.0	3.951	6.0	0.203	5.0	0.715	5.0	0.153	6.0	5.60
Computer	gmiGA-wrap	0.599	2.0	4.866	2.0	0.082	2.0	0.475	4.0	0.102	4.0	2.80
	gmi-ML-CFS	0.599	3.0	4.867	3.0	0.084	3.0	0.475	4.0	0.101	2.0	3.00
	NoFS	0.598	4.0	4.876	4.0	0.093	5.0	0.475	2.0	0.103	5.0	4.00
	BR(BPNN)	0.630	1.0	4.289	1.0	0.040	1.0	0.443	1.0	0.091	1.0	1.00
	CFS-U	0.594	6.0	4.876	5.0	0.089	4.0	0.475	4.0	0.104	6.0	5.00
	RFML	0.598	5.0	4.904	6.0	0.100	6.0	0.475	6.0	0.102	3.0	5.20
Science	gmiGA-wrap	0.396	5.0	7.842	5.0	0.129	4.0	0.758	4.0	0.157	5.0	4.60
	gmi-ML-CFS	0.397	4.0	7.747	4.0	0.124	2.0	0.758	4.0	0.156	4.0	3.60
	NoFS	0.393	6.0	7.873	6.0	0.212	6.0	0.758	4.0	0.158	6.0	5.60
	BR(BPNN)	0.456	1.0	6.852	1.0	0.035	1.0	0.676	1.0	0.134	1.0	1.00
	CFS-U	0.397	3.0	7.682	3.0	0.160	5.0	0.758	4.0	0.155	3.0	3.60
	RFML	0.400	2.0	7.582	2.0	0.125	3.0	0.758	4.0	0.153	2.0	2.60
MEAN	gmiGA-wrap		3.4		3.0		2.9		3.7		3.4	3.27
	gmi-ML-CFS		3.7		3.0		2.9		3.3		3.0	3.18
	NoFS		4.3		4.9		5.0		4.2		5.1	4.70
	BR(BPNN)		1.2		1.7		1.2		1.2		1.5	1.36
	CFS-U		3.7		3.7		4.4		3.9		3.7	3.87
	RFML		4.7		4.7		4.6		4.8		4.3	4.62

Table 6.30: Values of five multi-label predictive accuracy measures for BPMLL classifier with six different multi-label feature selection methods - feature space size = 200

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiGA-wrap	0.565	4.0	13.333	2.0	0.088	3.0	0.401	3.0	0.098	2.0	2.80
	gmi-ML-CFS	0.559	6.0	13.293	1.0	0.087	2.0	0.405	4.0	0.098	3.0	3.20
	NoFS	0.562	5.0	14.326	6.0	0.098	6.0	0.418	6.0	0.105	6.0	5.80
	BR(BPNN)	0.596	1.0	13.404	3.0	0.057	1.0	0.373	1.0	0.097	1.0	1.40
	CFS-U	0.572	2.0	13.969	5.0	0.092	5.0	0.409	5.0	0.102	5.0	4.40
	RFML	0.568	3.0	13.457	4.0	0.092	4.0	0.400	2.0	0.100	4.0	3.40
Medical	gmiGA-wrap	0.815	2.0	2.282	2.0	0.018	2.0	0.256	2.0	0.035	2.0	2.00
	gmi-ML-CFS	0.758	5.0	2.475	3.0	0.026	6.0	0.341	5.0	0.040	3.0	4.40
	NoFS	0.759	4.0	2.588	5.0	0.019	4.0	0.353	6.0	0.041	5.0	4.80
	BR(BPNN)	0.745	6.0	3.557	6.0	0.019	3.0	0.321	3.0	0.060	6.0	4.80
	CFS-U	0.836	1.0	2.200	1.0	0.014	1.0	0.219	1.0	0.033	1.0	1.00
	RFML	0.773	3.0	2.562	4.0	0.021	5.0	0.321	4.0	0.041	4.0	4.00
Business	gmiGA-wrap	0.854	3.0	2.715	3.0	0.043	6.0	0.139	5.0	0.048	2.0	3.80
	gmi-ML-CFS	0.853	6.0	2.751	6.0	0.041	4.0	0.139	2.0	0.049	4.0	4.40
	NoFS	0.853	5.0	2.728	5.0	0.041	2.0	0.139	5.0	0.049	5.0	4.40
	BR(BPNN)	0.876	1.0	2.299	1.0	0.028	1.0	0.124	1.0	0.041	1.0	1.00
	CFS-U	0.855	2.0	2.705	2.0	0.041	3.0	0.139	5.0	0.048	3.0	3.00
	RFML	0.854	4.0	2.723	4.0	0.042	5.0	0.139	3.0	0.049	6.0	4.40
Art	gmiGA-wrap	0.437	3.0	5.994	3.0	0.192	4.0	0.752	4.0	0.175	3.0	3.40
	gmi-ML-CFS	0.437	2.0	5.985	2.0	0.185	2.0	0.752	4.0	0.175	2.0	2.40
	NoFS	0.404	6.0	7.565	6.0	0.548	6.0	0.752	4.0	0.230	6.0	5.60
	BR(BPNN)	0.519	1.0	5.319	1.0	0.059	1.0	0.605	1.0	0.147	1.0	1.00
	CFS-U	0.428	5.0	6.184	5.0	0.287	5.0	0.752	4.0	0.183	5.0	4.80
	RFML	0.436	4.0	6.006	4.0	0.188	3.0	0.752	4.0	0.175	4.0	3.80
Education	gmiGA-wrap	0.481	2.0	4.505	3.0	0.124	2.0	0.681	2.0	0.107	2.0	2.20
	gmi-ML-CFS	0.480	3.0	4.477	2.0	0.142	4.0	0.681	4.5	0.107	3.0	3.30
	NoFS	0.469	6.0	5.298	6.0	0.261	6.0	0.681	4.5	0.122	6.0	5.70
	BR(BPNN)	0.544	1.0	3.895	1.0	0.041	1.0	0.602	1.0	0.092	1.0	1.00
	CFS-U	0.476	4.0	4.687	4.0	0.154	5.0	0.681	4.5	0.111	4.0	4.30
	RFML	0.474	5.0	4.757	5.0	0.134	3.0	0.681	4.5	0.112	5.0	4.50
Recreation	gmiGA-wrap	0.379	3.0	5.517	2.0	0.225	4.0	0.804	4.0	0.219	2.0	3.00
	gmi-ML-CFS	0.379	2.0	5.530	3.0	0.206	2.0	0.803	3.0	0.219	3.0	2.60
	NoFS	0.346	6.0	6.917	6.0	0.548	6.0	0.802	2.0	0.278	6.0	5.20
	BR(BPNN)	0.553	1.0	4.321	1.0	0.056	1.0	0.570	1.0	0.158	1.0	1.00
	CFS-U	0.370	5.0	5.939	5.0	0.386	5.0	0.805	5.5	0.237	5.0	5.10
	RFML	0.376	4.0	5.634	4.0	0.225	3.0	0.805	5.5	0.223	4.0	4.10
Health	gmiGA-wrap	0.616	3.0	3.934	2.0	0.111	2.0	0.488	2.0	0.077	3.0	2.40
	gmi-ML-CFS	0.617	2.0	3.976	3.0	0.113	3.0	0.489	5.0	0.077	2.0	3.00
	NoFS	0.606	6.0	4.148	6.0	0.158	6.0	0.489	5.0	0.082	6.0	5.80
	BR(BPNN)	0.673	1.0	3.453	1.0	0.044	1.0	0.412	1.0	0.065	1.0	1.00
	CFS-U	0.609	5.0	4.098	4.0	0.152	5.0	0.489	5.0	0.080	5.0	4.80
	RFML	0.614	4.0	4.121	5.0	0.131	4.0	0.489	3.0	0.079	4.0	4.00
Ent.ment	gmiGA-wrap	0.520	2.0	3.521	2.0	0.170	2.0	0.662	2.0	0.135	2.0	2.00
	gmi-ML-CFS	0.506	3.0	3.533	3.0	0.172	3.0	0.688	3.0	0.135	3.0	3.00
	NoFS	0.417	6.0	5.087	6.0	0.476	6.0	0.788	6.0	0.199	6.0	6.00
	BR(BPNN)	0.624	1.0	2.982	1.0	0.056	1.0	0.500	1.0	0.108	1.0	1.00
	CFS-U	0.480	4.0	3.799	4.0	0.266	5.0	0.715	5.0	0.149	4.0	4.40
	RFML	0.473	5.0	3.934	5.0	0.193	4.0	0.715	4.0	0.153	5.0	4.60
Computer	gmiGA-wrap	0.599	3.0	4.864	3.0	0.086	4.0	0.475	4.0	0.102	4.0	3.60
	gmi-ML-CFS	0.601	2.0	4.810	2.0	0.084	3.0	0.475	4.0	0.101	2.0	2.60
	NoFS	0.582	5.0	5.111	6.0	0.169	6.0	0.475	4.0	0.111	6.0	5.40
	BR(BPNN)	0.647	1.0	4.125	1.0	0.038	1.0	0.424	1.0	0.087	1.0	1.00
	CFS-U	0.570	6.0	5.087	5.0	0.114	5.0	0.475	4.0	0.110	5.0	5.00
	RFML	0.598	4.0	4.883	4.0	0.072	2.0	0.475	4.0	0.102	3.0	3.40
Science	gmiGA-wrap	0.397	3.0	7.741	3.0	0.136	4.0	0.758	4.0	0.156	3.0	3.40
	gmi-ML-CFS	0.396	4.0	7.811	4.0	0.129	2.0	0.758	4.0	0.157	4.0	3.60
	NoFS	0.382	6.0	9.138	6.0	0.478	6.0	0.758	4.0	0.188	6.0	5.60
	BR(BPNN)	0.476	1.0	6.617	1.0	0.034	1.0	0.654	1.0	0.129	1.0	1.00
	CFS-U	0.393	5.0	8.007	5.0	0.250	5.0	0.758	4.0	0.161	5.0	4.80
	RFML	0.399	2.0	7.645	2.0	0.134	3.0	0.758	4.0	0.154	2.0	2.60
MEAN	gmiGA-wrap		2.8		2.5		3.3		3.2		2.5	2.86
	gmi-ML-CFS		3.5		2.9		3.1		3.9		2.9	3.25
	NoFS		5.5		5.8		5.4		4.7		5.8	5.43
	BR(BPNN)		1.5		1.7		1.2		1.2		1.5	1.42
	CFS-U		3.9		4.0		4.4		4.3		4.2	4.16
	RFML		3.8		4.1		3.6		3.8		4.1	3.88

Table 6.31: Values of five multi-label predictive accuracy measures for BPMLL classifier with six different multi-label feature selection methods - feature space size = 300

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiGA-wrap	0.563	4.0	13.640	2.0	0.091	5.0	0.415	3.0	0.100	1.0	3.00
	gmi-ML-CFS	0.583	1.0	14.041	5.0	0.106	6.0	0.425	4.0	0.101	3.0	3.80
	NoFS	0.567	3.0	13.629	1.0	0.059	1.0	0.404	2.0	0.100	2.0	1.80
	BR(BPNN)	0.569	2.0	14.361	6.0	0.090	3.0	0.427	5.0	0.104	6.0	4.40
	CFS-U	0.561	5.0	13.708	3.0	0.091	4.0	0.400	1.0	0.101	4.0	3.40
	RFML	0.560	6.0	13.928	4.0	0.089	2.0	0.428	6.0	0.103	5.0	4.60
Medical	gmiGA-wrap	0.825	3.0	2.272	2.0	0.017	3.0	0.235	3.0	0.034	2.0	2.60
	gmi-ML-CFS	0.738	6.0	3.578	6.0	0.019	5.0	0.336	6.0	0.060	6.0	5.80
	NoFS	0.847	2.0	2.078	1.0	0.014	1.0	0.205	2.0	0.031	1.0	1.40
	BR(BPNN)	0.807	5.0	2.482	4.0	0.018	4.0	0.260	4.5	0.039	4.0	4.30
	CFS-U	0.854	1.0	2.735	5.0	0.038	6.0	0.139	1.0	0.048	5.0	3.60
	RFML	0.809	4.0	2.464	3.0	0.017	2.0	0.260	4.5	0.039	3.0	3.30
Business	gmiGA-wrap	0.853	4.0	2.762	4.0	0.034	1.0	0.139	2.5	0.049	3.0	2.90
	gmi-ML-CFS	0.853	3.0	2.757	3.0	0.038	3.0	0.139	2.5	0.049	4.0	3.10
	NoFS	0.854	1.0	2.720	2.0	0.042	4.0	0.139	2.5	0.048	1.0	2.10
	BR(BPNN)	0.436	6.0	6.000	6.0	0.204	6.0	0.752	5.5	0.175	6.0	5.90
	CFS-U	0.437	5.0	5.990	5.0	0.196	5.0	0.752	5.5	0.175	5.0	5.10
	RFML	0.853	2.0	2.715	1.0	0.037	2.0	0.139	2.5	0.048	2.0	1.90
Art	gmiGA-wrap	0.167	6.0	10.195	6.0	0.626	6.0	0.973	6.0	0.347	6.0	6.00
	gmi-ML-CFS	0.436	4.0	6.000	4.0	0.200	5.0	0.752	4.5	0.175	4.0	4.30
	NoFS	0.482	2.0	4.513	2.0	0.124	3.0	0.681	3.0	0.107	2.0	2.40
	BR(BPNN)	0.480	3.0	4.548	3.0	0.121	1.0	0.681	2.0	0.108	3.0	2.40
	CFS-U	0.482	1.0	4.474	1.0	0.122	2.0	0.678	1.0	0.106	1.0	1.20
	RFML	0.436	5.0	6.009	5.0	0.191	4.0	0.752	4.5	0.176	5.0	4.70
Education	gmiGA-wrap	0.541	1.0	3.914	1.0	0.041	1.0	0.604	1.0	0.092	1.0	1.00
	gmi-ML-CFS	0.376	4.0	5.702	5.0	0.327	5.0	0.804	4.0	0.226	5.0	4.60
	NoFS	0.376	5.0	5.661	4.0	0.254	4.0	0.805	5.0	0.224	4.0	4.40
	BR(BPNN)	0.379	3.0	5.561	3.0	0.222	3.0	0.802	3.0	0.220	3.0	3.00
	CFS-U	0.184	6.0	8.551	6.0	0.702	6.0	0.972	6.0	0.356	6.0	6.00
	RFML	0.474	2.0	4.757	2.0	0.144	2.0	0.681	2.0	0.112	2.0	2.00
Recreation	gmiGA-wrap	0.370	6.0	6.133	6.0	0.481	6.0	0.805	5.5	0.242	6.0	5.90
	gmi-ML-CFS	0.613	2.0	3.930	3.0	0.117	2.0	0.488	2.0	0.076	3.0	2.40
	NoFS	0.612	3.0	3.906	2.0	0.122	3.0	0.490	4.0	0.076	2.0	2.80
	BR(BPNN)	0.595	4.0	4.348	4.0	0.189	4.0	0.489	3.0	0.088	4.0	3.80
	CFS-U	0.674	1.0	3.441	1.0	0.045	1.0	0.418	1.0	0.065	1.0	1.00
	RFML	0.373	5.0	5.810	5.0	0.338	5.0	0.805	5.5	0.229	5.0	5.10
Health	gmiGA-wrap	0.607	2.0	4.094	4.0	0.132	2.0	0.489	1.5	0.081	2.0	2.30
	gmi-ML-CFS	0.529	4.0	3.455	2.0	0.154	4.0	0.649	4.0	0.132	4.0	3.60
	NoFS	0.217	6.0	7.002	6.0	0.532	6.0	0.951	6.0	0.304	6.0	6.00
	BR(BPNN)	0.608	1.0	3.034	1.0	0.057	1.0	0.523	3.0	0.111	3.0	1.80
	CFS-U	0.471	5.0	4.054	3.0	0.314	5.0	0.715	5.0	0.157	5.0	4.60
	RFML	0.606	3.0	4.104	5.0	0.132	3.0	0.489	1.5	0.081	1.0	2.70
Ent.ment	gmiGA-wrap	0.598	2.0	4.914	3.0	0.091	3.0	0.475	3.5	0.104	2.0	2.70
	gmi-ML-CFS	0.235	6.0	8.556	6.0	0.475	6.0	0.971	6.0	0.211	6.0	6.00
	NoFS	0.651	1.0	4.086	2.0	0.037	1.0	0.423	1.0	0.086	1.0	1.20
	BR(BPNN)	0.588	4.0	5.205	5.0	0.207	5.0	0.475	2.0	0.111	4.0	4.00
	CFS-U	0.597	3.0	4.965	4.0	0.082	2.0	0.475	3.5	0.105	3.0	3.10
	RFML	0.473	5.0	3.925	1.0	0.188	4.0	0.715	5.0	0.153	5.0	4.00
Computer	gmiGA-wrap	0.396	5.0	7.790	5.0	0.133	3.0	0.758	5.0	0.157	5.0	4.60
	gmi-ML-CFS	0.475	3.0	6.611	3.0	0.034	1.0	0.660	3.0	0.130	3.0	2.60
	NoFS	0.388	6.0	8.727	6.0	0.453	6.0	0.758	5.0	0.177	6.0	5.80
	BR(BPNN)	0.398	4.0	7.656	4.0	0.137	4.0	0.758	5.0	0.155	4.0	4.20
	CFS-U	0.588	2.0	5.205	2.0	0.207	5.0	0.475	1.0	0.111	2.0	2.40
	RFML	0.598	1.0	4.908	1.0	0.080	2.0	0.475	2.0	0.104	1.0	1.40
Science	gmiGA-wrap	0.395	4.0	7.987	4.0	0.182	4.0	0.758	3.5	0.161	4.0	3.90
	gmi-ML-CFS	0.396	3.0	7.815	3.0	0.129	2.0	0.758	3.5	0.157	3.0	2.90
	NoFS	0.153	6.0	12.225	6.0	0.546	6.0	0.981	6.0	0.268	6.0	6.00
	BR(BPNN)	0.475	1.0	6.611	1.0	0.034	1.0	0.660	1.0	0.130	1.0	1.00
	CFS-U	0.388	5.0	8.727	5.0	0.453	5.0	0.758	3.5	0.177	5.0	4.70
	RFML	0.398	2.0	7.698	2.0	0.142	3.0	0.758	3.5	0.156	2.0	2.50
MEAN	gmiGA-wrap		3.7		3.7		3.4		3.5		3.2	3.49
	gmi-ML-CFS		3.6		4.0		3.9		4.0		4.1	3.91
	NoFS		3.5		3.2		3.5		3.7		3.1	3.39
	BR(BPNN)		3.3		3.7		3.2		3.4		3.8	3.48
	CFS-U		3.4		3.5		4.1		2.9		3.7	3.51
	RFML		3.5		2.9		2.9		3.7		3.1	3.22

Table 6.32: Values of five multi-label predictive accuracy measures for BPMLL classifier with six different multi-label feature selection methods - feature space size = 400

Dataset	Methods	Predictive Accuracy Measures and Ranking										
		Avg-Pre	R	Coverage	R	H-Loss	R	OneError	R	R-Loss	R	AR
Enron	gmiGA-wrap	0.566	2.0	13.995	3.0	0.092	3.0	0.428	3.0	0.102	3.0	2.80
	gmi-ML-CFS	0.559	3.0	13.188	1.0	0.089	2.0	0.396	2.0	0.097	1.0	1.80
	NoFS	0.553	5.0	14.663	5.0	0.124	6.0	0.431	4.0	0.111	5.0	5.00
	BR(BPNN)	0.583	1.0	13.397	2.0	0.056	1.0	0.382	1.0	0.098	2.0	1.40
	CFS-U	0.552	6.0	14.828	6.0	0.096	5.0	0.435	6.0	0.112	6.0	5.80
	RFML	0.557	4.0	14.282	4.0	0.094	4.0	0.434	5.0	0.105	4.0	4.20
Medical	gmiGA-wrap	0.809	2.0	2.321	3.0	0.017	2.0	0.267	2.0	0.035	3.0	2.40
	gmi-ML-CFS	0.795	3.0	2.504	4.0	0.019	4.0	0.276	3.0	0.040	4.0	3.60
	NoFS	0.154	6.0	14.135	6.0	0.325	6.0	0.940	6.0	0.292	6.0	6.00
	BR(BPNN)	0.728	5.0	3.716	5.0	0.020	5.0	0.349	5.0	0.063	5.0	5.00
	CFS-U	0.788	4.0	2.196	1.0	0.017	3.0	0.318	4.0	0.033	1.0	2.60
	RFML	0.835	1.0	2.206	2.0	0.015	1.0	0.217	1.0	0.033	2.0	1.40
Business	gmiGA-wrap	0.858	2.0	2.630	2.0	0.042	5.0	0.139	3.5	0.046	2.0	2.90
	gmi-ML-CFS	0.849	5.0	2.804	5.0	0.039	3.0	0.139	3.5	0.050	5.0	4.30
	NoFS	0.579	6.0	4.664	6.0	0.349	6.0	0.475	6.0	0.100	6.0	6.00
	BR(BPNN)	0.881	1.0	2.258	1.0	0.028	1.0	0.119	1.0	0.039	1.0	1.00
	CFS-U	0.856	3.0	2.646	3.0	0.041	4.0	0.139	3.5	0.047	3.0	3.30
	RFML	0.853	4.0	2.734	4.0	0.035	2.0	0.139	3.5	0.049	4.0	3.50
Art	gmiGA-wrap	0.432	4.0	6.209	4.0	0.319	4.0	0.752	3.0	0.183	4.0	3.80
	gmi-ML-CFS	0.436	2.0	6.000	2.0	0.197	3.0	0.752	3.0	0.175	2.0	2.40
	NoFS	0.151	6.0	11.617	6.0	0.460	5.0	0.984	6.0	0.397	6.0	5.80
	BR(BPNN)	0.509	1.0	5.342	1.0	0.060	1.0	0.631	1.0	0.147	1.0	1.00
	CFS-U	0.337	5.0	8.150	5.0	0.544	6.0	0.843	5.0	0.257	5.0	5.20
	RFML	0.436	3.0	6.016	3.0	0.195	2.0	0.752	3.0	0.176	3.0	2.80
Education	gmiGA-wrap	0.479	2.0	4.616	2.0	0.140	3.0	0.681	3.5	0.109	2.0	2.50
	gmi-ML-CFS	0.476	3.0	4.689	3.0	0.131	2.0	0.681	3.5	0.111	3.0	2.90
	NoFS	0.121	6.0	11.883	6.0	0.497	6.0	0.987	6.0	0.342	6.0	6.00
	BR(BPNN)	0.535	1.0	3.950	1.0	0.042	1.0	0.611	1.0	0.093	1.0	1.00
	CFS-U	0.470	4.0	5.194	5.0	0.233	5.0	0.681	3.5	0.120	4.0	4.30
	RFML	0.470	5.0	5.085	4.0	0.172	4.0	0.681	3.5	0.120	5.0	4.30
Recreation	gmiGA-wrap	0.374	3.0	5.831	4.0	0.368	4.0	0.805	3.0	0.232	4.0	3.60
	gmi-ML-CFS	0.373	4.0	5.818	3.0	0.330	2.0	0.805	3.0	0.229	2.0	2.80
	NoFS	0.159	6.0	10.782	6.0	0.567	6.0	0.975	6.0	0.447	6.0	6.00
	BR(BPNN)	0.552	1.0	4.238	1.0	0.057	1.0	0.576	1.0	0.155	1.0	1.00
	CFS-U	0.334	5.0	6.674	5.0	0.547	5.0	0.840	5.0	0.270	5.0	5.00
	RFML	0.375	2.0	5.812	2.0	0.352	3.0	0.805	3.0	0.230	3.0	2.60
Health	gmiGA-wrap	0.618	2.0	3.814	2.0	0.118	3.0	0.489	4.0	0.073	2.0	2.60
	gmi-ML-CFS	0.617	3.0	3.848	3.0	0.116	2.0	0.489	2.0	0.074	3.0	2.60
	NoFS	0.308	6.0	7.135	6.0	0.404	6.0	0.883	6.0	0.173	6.0	6.00
	BR(BPNN)	0.692	1.0	3.303	1.0	0.043	1.0	0.395	1.0	0.061	1.0	1.00
	CFS-U	0.587	5.0	4.688	5.0	0.217	5.0	0.489	4.0	0.096	5.0	4.80
	RFML	0.605	4.0	4.126	4.0	0.135	4.0	0.489	4.0	0.081	4.0	4.00
Ent.ment	gmiGA-wrap	0.469	4.0	4.021	4.0	0.301	4.0	0.715	5.0	0.157	4.0	4.20
	gmi-ML-CFS	0.498	2.0	3.589	2.0	0.189	2.0	0.705	2.0	0.139	2.0	2.00
	NoFS	0.202	6.0	7.131	6.0	0.576	6.0	0.974	6.0	0.310	6.0	6.00
	BR(BPNN)	0.617	1.0	2.997	1.0	0.057	1.0	0.510	1.0	0.110	1.0	1.00
	CFS-U	0.461	5.0	4.371	5.0	0.367	5.0	0.715	3.0	0.169	5.0	4.60
	RFML	0.473	3.0	3.929	3.0	0.211	3.0	0.715	4.0	0.153	3.0	3.20
Computer	gmiGA-wrap	0.583	4.0	5.131	4.0	0.126	4.0	0.475	3.0	0.110	4.0	3.80
	gmi-ML-CFS	0.596	3.0	4.980	3.0	0.086	3.0	0.475	3.0	0.106	3.0	3.00
	NoFS	0.135	6.0	11.156	6.0	0.574	6.0	0.983	6.0	0.301	6.0	6.00
	BR(BPNN)	0.655	1.0	4.030	1.0	0.037	1.0	0.418	1.0	0.084	1.0	1.00
	CFS-U	0.363	5.0	7.035	5.0	0.451	5.0	0.848	5.0	0.158	5.0	5.00
	RFML	0.597	2.0	4.947	2.0	0.084	2.0	0.475	3.0	0.105	2.0	2.20
Science	gmiGA-wrap	0.393	4.0	8.133	4.0	0.235	4.0	0.758	3.0	0.165	4.0	3.80
	gmi-ML-CFS	0.396	2.0	7.787	2.0	0.129	2.0	0.758	3.0	0.157	2.0	2.20
	NoFS	0.128	6.0	14.598	6.0	0.592	6.0	0.980	6.0	0.329	6.0	6.00
	BR(BPNN)	0.462	1.0	6.680	1.0	0.035	1.0	0.671	1.0	0.132	1.0	1.00
	CFS-U	0.269	5.0	10.384	5.0	0.489	5.0	0.893	5.0	0.219	5.0	5.00
	RFML	0.395	3.0	7.942	3.0	0.190	3.0	0.758	3.0	0.162	3.0	3.00
MEAN	gmiGA-wrap		2.9		3.2		3.6		3.3		3.2	3.24
	gmi-ML-CFS		3.0		2.8		2.5		2.8		2.7	2.76
	NoFS		5.9		5.9		5.9		5.8		5.9	5.88
	BR(BPNN)		1.4		1.5		1.4		1.4		1.5	1.44
	CFS-U		4.7		4.5		4.8		4.4		4.4	4.56
	RFML		3.1		3.1		2.8		3.3		3.3	3.12

Table 6.33: Summary of results in terms of average ranking (AR) and the number of selected features (Sel.F) of gmiGA-wrap and other multi-label feature selection methods using BPMLL as classifier

FS. size	NoFS		BR(BPNN)		CFS-U		RFML		gmi-ML-CFS		gmiGA-wrap	
	AR	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**	**AR**	**Sel.F**
100	4.70	100.00	**1.36**	100.00	3.87	73.90	4.62	25.76	3.18	22.40	3.27	25.76
200	5.43	200.00	**1.42**	200.00	4.16	128.40	3.88	50.74	3.25	34.30	2.86	50.74
300	3.39	300.00	3.48	300.00	3.51	174.80	**3.22**	80.90	3.91	44.10	3.49	80.90
400	5.88	400.00	**1.44**	400.00	4.56	214.40	3.12	116.50	2.76	57.00	3.24	116.50
Avg	4.85	250.00	**1.93**	250.00	4.03	147.88	3.71	68.48	3.28	39.45	3.22	68.48

Table 6.34: Summary of overall average ranking (AR) across four individual lengths for four versions of gmiGA-wrap and other multi-label feature selection methods using BPMLL as the classifier

Dataset	Overall Average Rank (AR) across 4 individual lengths					
	gmiGA-wrap	**gmi-ML-CFS**	**NoFS**	**BR(BPNN)**	**CFS-U**	**RFML**
Enron	3.00(3)	2.95(2)	4.45(6)	2.20(1)	4.40(5)	4.00(4)
Medical	2.55(2)	4.90(6)	3.60(4)	4.53(5)	2.10(1)	3.33(3)
Business	3.38(2)	3.55(3)	3.90(5)	2.23(1)	4.08(6)	3.88(4)
Art	4.15(5)	3.33(2)	4.85(6)	1.35(1)	3.55(3)	3.78(4)
Education	2.13(2)	3.35(3)	5.33(6)	1.50(1)	4.70(5)	4.00(4)
Recreation	3.68(3)	2.75(2)	4.93(6)	1.70(1)	3.75(4)	4.20(5)
Health	2.58(2)	2.80(3)	5.70(6)	1.20(1)	4.88(5)	3.85(4)
Ent.ment	3.08(2)	3.25(3)	4.65(6)	1.75(1)	3.93(4)	4.35(5)
Computer	3.70(4)	2.80(2)	5.30(6)	1.80(1)	4.35(5)	3.05(3)
Science	3.93(4)	3.08(3)	5.80(6)	1.00(1)	4.53(5)	2.68(2)
Average	3.22(2.9)	3.28(2.9)	4.85(5.7)	**1.93(1.4)**	4.03(4.3)	3.71(3.8)

other multi-label feature selection methods plotted against the average number of selected features across all datasets and feature space sizes, when using BPMLL as the classifier. Again, clearly, gmiGA-wrap outperforms all other methods in terms of selecting a smaller number of features. In addition, gmiML-CFS achieves a reasonable average ranking, although clearly worse than BR(BPNN). However, BR(BPNN) did not achieve a good trade-off between average ranking and number of selected features, as shown in Figure 6.4

Table 6.34 shows the overall average rank of six multi-label feature selection methods for each dataset (averaged across the 4 GA individual lengths). The first value in each cell is the actual average rank, whilst the value between brackets is the "rank of the average rank". This later value was used in the Friedman

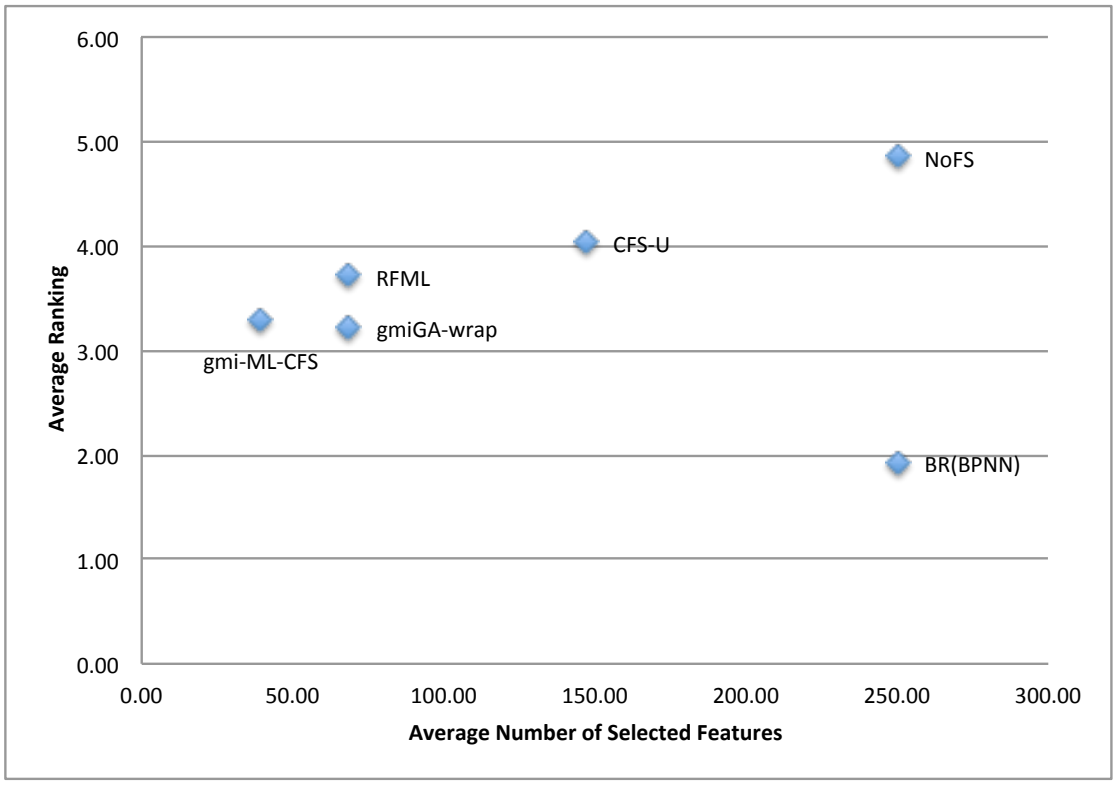


Figure 6.4: Overall average ranking (AR) for gmiGA-wrap and the other multi-label feature selection methods plotted against the average number of selected features across all datasets and feature space sizes, when using BPMLL as the classifier

and Holm's test (as discussed at the end of Subsection 6.8.2). We confidently conclude that there is a significant difference among the 6 algorithms on 10 evaluation datasets at the 0.05 significance level for a two tailed test.

Then, the Holm's posthoc test was applied on these data using gmiGA-wrap as the control method. There is a significant difference between gmiGA-wrap and NoFS at the 0.05 significant level (p value = 0.01063) but there are no significance differences between gmiGA-wrap and the other 5 methods at the same level of significance.

In addition, we also compared the time taken to run the BR approach (using all input features), which obtained the best average rank with the BPMLL classi-

fier, versus the computational time taken to first run GA-ML-CFS and then run BPMLL using only the features selected by GA-ML-CFS.

Table 6.35 shows GA-ML-CFS' running time on three multi-label evaluation datasets. These datasets were selected based on their number of labels. The Enron dataset has the largest number of labels (53 labels) among the 10 evaluation datasets, the Entertainment dataset has the smallest number of labels (21 labels) and the Health dataset has an intermediate number of labels (32 labels). For each of those datasets, the table reports the running time for 4 versions of the datasets, varying the number of input features from 100 to 400.

In this Table, the second column shows the running time of GA-ML-CFS, the third column shows the time for running BPMLL using only the features selected by GA-ML-CFS, the fourth column shows the summation of the previous two running times, and the last column shows the running time of the BR approach, using all input features. Note that all experiments which measure the computational time were run on a system with Intel Core i7 CPU at 3.40 GHz and 16.0 GB of memory. Also, the running time format (d:h:m:s) shown in Table 6.35 refers to days, hours, minutes and seconds.

The Entertainment dataset has the longest time for running GA-ML-CFS (more than 5 days with 400 input features) because the size of this dataset is relatively large in terms of the number of instances (12,730 instances) while the Enron dataset took a relatively short time to process because this dataset has a relatively small number of instances (1,702 instances). Clearly, BPMLL's running time is very small when using only the features selected by GA-ML-CFS. The longest BPMLL's running time is 1 minute and 18 seconds on the biggest dataset, Entertainment 400. Moreover, the running time of the BR approach, which used the full set of input features, obviously takes much more time than the GA-ML-CFS' running time plus BPMLL's running time in most cases. The exceptions are

Table 6.35: Comparing the computational time of GA-ML-CFS and BPMLL versus the BR approach on three different datasets

Dataset	Running time (d:h:m:s)			
	GA-ML-CFS	**BPMLL with feature selected by GA-ML-CFS**	**BPMLL and GA-ML-CFS**	**BR with BP using full set of features**
Enron 100	0:01:15:12	0:00:00:08	0:01:15:20	0:04:21:51
Enron 200	0:03:21:41	0:00:00:08	0:03:21:49	0:11:20:00
Enron 300	0:06:23:48	0:00:00:13	0:06:24:01	1:04:03:55
Enron 400	0:12:23:50	0:00:00:15	0:12:24:05	2:08:53:10
Entertainment 100	0:08:48:58	0:00:00:25	0:08:49:22	0:06:56:50
Entertainment 200	1:08:11:34	0:00:00:40	1:08:12:14	1:10:14:10
Entertainment 300	2:17:43:00	0:00:00:56	2:17:43:56	2:08:20:15
Entertainment 400	5:06:19:08	0:00:01:18	5:06:19:28	4:19:12:05
Health 100	0:02:44:12	0:00:00:20	0:02:44:32	0:13:30:00
Health 200	0:08:24:00	0:00:00:20	0:08:24:20	1:10:14:10
Health 300	0:20:30:13	0:00:00:26	0:20:30:39	2:17:30:56
Health 400	1:10:59:35	0:00:00:37	1:11:00:12	5:10:12:28

the Entertainment datasets, where the difference of computational time between the two approaches is not large, and where in 3 of 4 cases (with 100, 300 and 400 input features) the BR approach took somewhat less time than the time to run both GA-ML-CFS and BPMLL with the selected features.

6.9 Conclusion

This Chapter proposed two versions of the new Genetic Algorithm for Multi-Label Correlation-Based Feature Selection (GA-ML-CFS) method; one version using a single-objective fitness function, described in Section 6.2, and another version based on lexicographic multi-objective optimization, described in Section 6.3.

The first version of GA-ML-CFS proposed in this Chapter extends our previous version of ML-CFS (proposed in Chapter 4) by replacing the simple greedy strategy by a more sophisticated GA as a search method. The GA uses the genetic operators of crossover and mutation and a fitness-based selection method to explore the space of candidate feature subsets.

The second version of GA-ML-CFS, based on the lexicographic multi-objective approach, assigns different priorities to different objectives (evaluation criteria), and then focuses on optimizing the objectives in decreasing order of priority. In our case, the highest priority objective was to maximize the value of the Merit of a feature subset, whilst the lowest priority was to minimize the number of selected features.

For each of the two versions of GA-ML-CFS, we tried two approaches for optimizing its parameter settings: a "wrapper-like" approach and a filter approach. We compared the predictive accuracy associated with four methods: two GA-ML-CFS versions times two parameter optimization approaches, in experiments using two well-known multi-label classification algorithms: ML-kNN and BPMLL as the multi-label classifier.

In general, the single-objective version of GA-ML-CFS with parameter optimized by the wrapper-like approach (gmiGA-wrap) obtained the best results. Hence, next we ran experiments with gmiGA-wrap and other multi-label feature selection methods to compare the predictive accuracy associated with their selected features again using ML-kNN and BPMLL. From the experimental results reported in this Chapter, in general when using MLkNN as classifier gmiGA-wrap obtained the second best predictive accuracy, and it clearly outperformed gmiML-CFS, NoFS, BR and RFML. In addition, gmiML-CFS selected the smallest feature subset but obtained the fourth best accuracy (out of 6 methods). The best predictive accuracy was obtained by CFS-U, but there was no statistically significant difference between the results of gmiGA-wrap and CFS-U. In addition, CFS-U selects on average about twice as many features as gmiGA-wrap.

When using BPMLL as the classifier, gmiGA-wrap obtained the second best predictive accuracy, although this time there is a very small difference between the

average ranks of gmiGA-wrap and gmiML-CFS (the third best method regarding accuracy), as shown in the last rows of Table 6.33 and 6.34. gmiML-CFS again selected on average the smallest feature subset. The best predictive accuracy was obtained by BR (Binary Relevance), but there was no statistically significant difference between the results of BR and gmiGA-wrap. In addition, BR does not reduce the number of features, since it uses all input features.

Chapter 7

Conclusions and Future Work

In this thesis we have focused on multi-label feature selection methods for multi-label classification problems. At the beginning, we proposed the first version of our multi-label correlation-based feature selection method (ML-CFS), which extended the well known single-label correlation-based feature selection (CFS) to the multi-label scenario. After that, we continued to improve ML-CFS in different dimensions, as described in Section 7.1.

In addition, we also proposed a new approach for using the single-label CFS method in a multi-label classification scenario. This approach first applies the single-label CFS method to variations of the original dataset containing all features and each class label separately; and then returns, as the selected feature subset, the union of the feature subsets selected by the separate applications of the single-label CFS method. This approach was called CFS-U, where U stands for the union of feature subsets for all class labels. In the remainder of this chapter, however, we focus on the proposed multi-label versions of the CFS method, which directly cope with multi-label classification datasets in a single run of the method.

We used multi-label datasets obtained from the MULAN repository for the experiments reported in Chapters 4 and 6. In Chapter 5, where we proposed and

evaluated ML-CFS versions exploiting biological knowledge, we have used two multi-label microarray gene expression datasets, which are not publically available. These datasets were prepared for data mining by the author of this thesis, using data provided by Prof. Michaelis, School of Bioscience, University of Kent.

In Chapters 4 - 6, two well-known multi-label classification algorithms, namely the Multi-Label k-Nearest Neighbour (ML-kNN) classification algorithm [124] and the Back-Propagation Multi-Label Learning (BPMLL) classification algorithm [123] were used to evaluate the quality of the feature subsets selected by all ML-CFS versions. That is, the features selected by ML-CFS were used as input by ML-kNN and BPMLL, and then the predictive accuracy of each classification model was measured, for each ML-CFS version, on the test set, containing data instances which were not included in the training set, therefore measuring the generalization ability of the classification model. Note that we measured the predictive accuracy using five different accuracy measures, namely: Hamming-loss, Ranking-loss, One-error, Coverage and Average Precision [113], as reviewed in Chapter 2. We also computed the average rank of each ML-CFS version across all accuracy measures and all datasets used in each experiment, and the overall results mentioned later in this Chapter refer to such average ranks.

We constructed the structure of our experiments and our thesis into three main parts: (1) the proposed ML-CFS methods based on hill climbing search, discussed in Chapter 4; (2) the proposed ML-CFS methods exploiting biological knowledge, discussed in Chapter 5; and (3) the proposed ML-CFS methods based on evolutionary algorithms, discussed in Chapter 6. The summary of contributions of this thesis is presented in Section 7.1, and the discussion of future research directions is presented in Section 7.2.

7.1 Summary of Contributions

As mentioned earlier, this thesis has proposed three types of ML-CFS methods, each having different versions, as shown in Figure 7.1. The meaning of the acronyms and method names in this Figure can be found in the corresponding Section where they were presented, which is indicated between brackets in the corresponding node in the Figure. Next, we summarize the main contributions in terms of new ML-CFS methods, with one Section for each of the three types of ML-CFS methods shown in Figure 7.1.

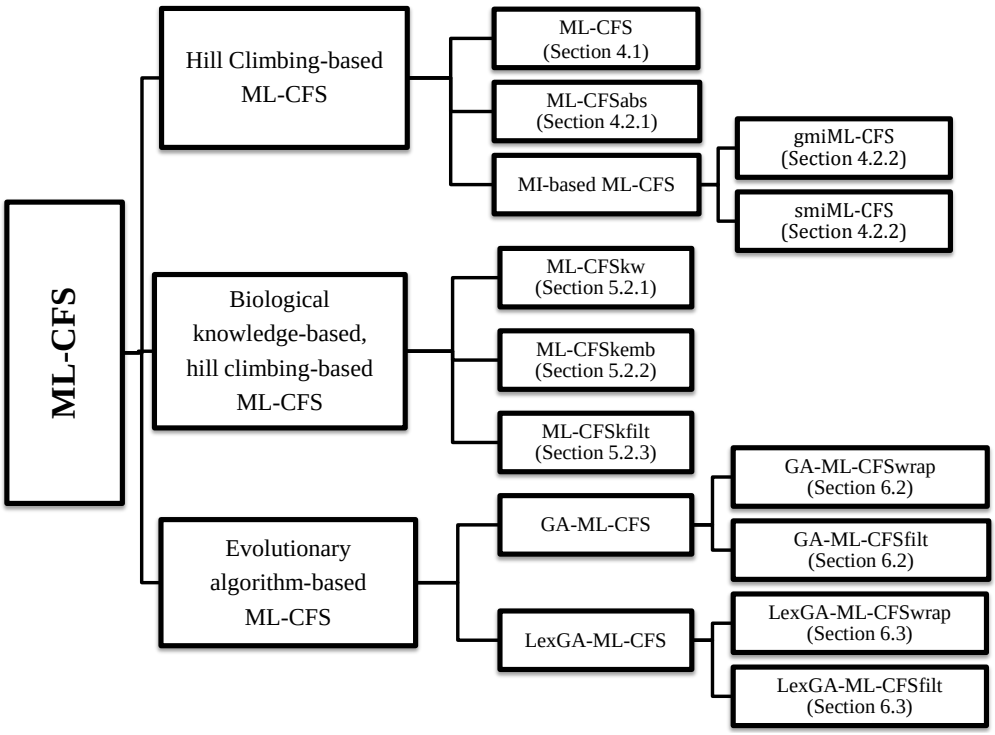


Figure 7.1: Summary of Original Contributions: ML-CFS methods

7.1.1 Multi-Label Correlation-Based Feature Selection (ML-CFS) Methods Based on Hill Climbing Search

Three extensions of ML-CFS were proposed: (1) The First Version of ML-CFS, (2) ML-CFS with the Absolute Value of Correlation Coefficient, and (3) ML-CFS using Mutual Information for Class Label Weighting. More precisely the third extension has two versions named gmiML-CFS and smiML-CFS. Note that gmiML-CFS stands for the ML-CFS version where class labels with greater MI (Mutual Information) are assigned greater weights, while smiML-CFS stands for the ML-CFS version where class labels with smaller MI are assigned greater weights.

7.1.1.1 The First Version of the ML-CFS Method

The first version of the ML-CFS method was proposed in [57]. This method extended the single-label CFS method [44] to multi-label classification problems. In general, ML-CFS uses a heuristic merit function to evaluate the merit of candidate feature subsets (like in single-label CFS). The difference between these methods is that the merit function of ML-CFS computes the average correlation coefficient between each feature in a candidate feature subset and each of the multiple class labels. By contrast, in the conventional single-label CFS method the merit function is simpler, because there is no need to measure average correlations over multiple class labels. The preliminary results of ML-CFS were discussed in [57], while the computational results of ML-CFS on other datasets were shown in Chapter 4.

7.1.1.2 ML-CFS with the Absolute Value of Correlation Coefficient

This approach improves the performance of the original ML-CFS using the property of the absolute value. In the first version of the multi-label ML-CFS method [57], Pearson's correlation coefficient (r) was used to estimate the correlation between features and labels, and the correlation between pair of features in a candi-

date feature subset. Note that both positive correlation and negative correlation can represent redundancy between a pair of features, or represent the relevance of a feature to predict a set of labels. However, in the original single-label and multi-label CFS methods, the value of the merit function depends on both the value and the sign of r. As discussed earlier, negative and positive values of correlation could cancel each other and produce a misleading merit value. Hence, the absolute (without sign) value of the correlation coefficient was used in all occurrences of the correlation coefficient in the merit function of this new ML-CFS version. The computational results in Section 4.4.2 and 4.4.3 show that ML-CFS with the absolute value of the correlation coefficient substantially improved the performance of ML-CFS on all evaluation datasets when using ML-kNN and BPMLL classifiers. Hence, the absolute value of the correlation coefficient was also used in all subsequent versions of ML-CFS presented in this thesis.

7.1.1.3 ML-CFS using Mutual Information for Class Label Weighting

The idea of this approach is that, when there are strong dependences among labels in the data, simply ignoring label correlations may not be sufficient to cope well with the label-dependence problem. To take label dependences into account, we used mutual information (MI) to measure the dependency between each pair of labels. We use MI, rather than Pearson's correlation coefficient, because labels are nominal, rather than numerical, and MI is often used to measure dependencies between nominal variables in feature selection. We proposed two MI-based ML-CFS versions. From the experimental results reported in Sections 4.4.4 and 4.4.5, one ML-CFS version using MI for class label weighting clearly outperforms the previous two versions of ML-CFS in general. Moreover, when comparing ML-CFS using MI for class label weighting with other multi-label feature selection methods, this method still shows a good predictive performance (it obtained the second best predictive accuracy out of five feature selection approaches) when using ML-kNN and BPMLL classifiers. In addition, gmiML-CFS selects substantially smaller fea-

ture subsets than the method that obtained the best predictive accuracy for each classifier.

7.1.2 Multi-Label Correlation-Based Feature Selection (ML-CFS) Methods that Exploit Biological Knowledge

Three extensions of ML-CFS that exploit biological knowledge were proposed: (1) ML-CFS using a Weighted Formula to Combine the Merit Function and KEGG Pathway Information, (2) ML-CFS Embedding KEGG Pathway Information into the Merit Function, and (3) ML-CFS Embedding KEGG Pathway Information into the Merit Function.

7.1.2.1 ML-CFS using a Weighted Formula to Combine the Merit Function and KEGG Pathway Information

In this approach we extended the ML-CFS method's evaluation function to use some biological knowledge about cancer-related pathways, to try to improve the predictive performance of ML-CFS and select genes (features) whose role in cancer-related drug resistance/sensitivity is more likely to be meaningful to biologists. We assumed that if some genes are related with cancer-related drug resistance/sensitivity to anti-cancer drugs, they are likely to occur in some cancer-related pathway(s). A set of these pathways were identified by Prof. Martin Michaelis, School of Biosciences. In order to quantify the strength of the relationship between the genes (features) in a candidate feature subset and the aforementioned cancer-related pathways, we proposed to compute "the average relative frequency of pathways per gene" as discussed earlier. This measure was then used as one of the terms in a new formula to measure a feature subset quality, where the other term was the merit function. Each term was assigned a numerical weight.

We ran experiments comparing 5 different weight settings of ML-CFS using a weighted formula to combine the merit function and frequency of pathway information. Clearly, the ML-CFS version with a weight of 0.9 for merit function and weight 0.1 for "the average relative frequency of pathways per gene", called ML-CFSk91, outperformed other versions of ML-CFS using a weighted formula on two gene expression datasets prepared as part of this research, as mentioned earlier. The details of these computational results are shown in Subsection 5.5.2. In addition, ML-CFSk91 also outperformed two previous versions of ML-CFS that do not exploit biological knowledge, as discussed in Subsection 5.5.3.

7.1.2.2 ML-CFS Embedding KEGG Pathway Information into the Merit Function

In this approach, we embedded the value of the "average relative frequency of pathways per gene" into the merit function in order to avoid the need to specify user-defined weights in our evaluation function (as in the previous ML-CFS version). In this approach, the formula to calculate the average value of the correlation between all features in a feature subset and all the labels in the class label set was extended to reward the feature-label correlation values in proportion to the strength of the association between the genes (features) in a candidate feature subset and pre-identified relevant cancer-related pathways.

We ran an experiment for comparing this new ML-CFS version with the previous ML-CFS version using a weighted formula to combine the merit function and cancer-related pathway information, on two microarray datasets. Clearly, ML-CFSk91 outperformed ML-CFS embedding KEGG Pathway information into the merit function on the two microarray datasets. The details of these computational results are shown in Section 5.5.4.

7.1.2.3 ML-CFS on datasets with pre-selected cancer-related features

The idea behind this approach was to investigate what would happen if we forced our feature selection method (ML-CFS) to select a feature subset from a feature space containing only the genes (features) that occur in some cancer-related pathway. Hence, in this approach we removed all genes which do not occur in any cancer-related pathway from the feature space. After that we gave all the remaining genes (i.e. all the genes occurring in some cancer-related pathway) as input to the ML-CFS method.

We ran an experiment for comparing this ML-CFS version against the previous two ML-CFS versions exploiting biological knowledge. Clearly, again ML-CFSk91 outperformed the other two ML-CFS versions on the two microarray datasets. The details of these computational results are shown in Section 5.5.4.

7.1.3 Multi-Label Correlation-Based Feature Selection (ML-CFS) Methods Based on Evolutionary Algorithms

Two new Genetic Algorithms for Multi-Label Correlation-Based Feature Selection (GA-ML-CFS) were proposed: (1) A Genetic Algorithm for ML-CFS using a single-objective fitness function, and (2) another version based on lexicographic multi-objective optimization named Lexicographic Genetic Algorithm for ML-CFS.

7.1.3.1 A Genetic Algorithm for ML-CFS

The new Genetic Algorithm for Multi-Label Correlation-Based Feature Selection (GA-ML-CFS) extends our previous versions of ML-CFS (described in [57, 58, 59]) by replacing the simple greedy strategy by a more sophisticated genetic algorithm

as a search method. The GA's fitness function was initially designed as the same merit function used by the greedy ML-CFS. Recall that a GA performs a more global search in the feature space than a greedy search because a GA works with a population of candidate solutions spread across different regions of the search space. As a result of their global search, GAs cope better with feature interaction and are less likely to get trapped into a local optimum in the search space, being more likely to find a global optimum. This new GA-ML-CFS method was presented in [56].

Moreover, we improved GA-ML-CFS' fitness function. According to the computational results reported in Section 4.4, clearly, ML-CFS using mutual information for class label weighting outperformed other ML-CFS versions and other multi-label feature selection methods. Hence, we decided to extend the fitness function originally based on the merit formula only, to consider also the mutual information for class label weighting. We also run experiments to find the recommended parameter setting for GA-ML-CFS using two different approaches: a "wrapper-like" approach and a filter approach. As shown in Section 6.7, clearly, GA-ML-CFS with parameter settings optimized by the wrapper-like approach (gmiGA-wrap) obtained the best results.

7.1.3.2 A Lexicographic Genetic Algorithm for ML-CFS

Lexicographic multi-objective optimization is a type of optimization technique which assigns different priorities to different objectives and optimizes each of the objectives in order of their priority. If one solution is significantly better than another with respect to the first criterion, this solution will be chosen. Otherwise, the performance of the two solutions is compared using the second criterion. LexGA-ML-CFS was proposed in [55]. In LexGA-ML-CFS, the fitness of an individual is evaluated based on two criteria (objectives): (1) the merit function (highest priority); and (2) the number of selected features (k) (lowest priority). Also, a

lexicographic optimization tournament selection was used in the parent selection step of the GA.

Similarly to the single-objective GA-ML-CFS, we improved the performance of LexGA-ML-CFS using the mutual information for class label weighting and we also ran experiments to find the recommended parameter settings for LexGA-ML-CFS using both a "wrapper-like" approach and a filter approach. The computational results comparing LexGA-ML-CFS with the single-objective GA-ML-CFS are shown in Section 6.7, where the single-objective GA-ML-CFS with parameter optimized by the wrapper-like approach (gmiGA-wrap) obtained the best results.

Finally, as reported in Sections 6.8.2 and 6.8.3, the best version of the proposed ML-CFS methods, namely GA-ML-CFS (with a single objective), obtain the second best predictive accuracy among 6 feature selection approaches being compared using both ML-kNN and BPMLL as classifiers. However, there was no statistically significant difference between the results obtained by GA-ML-CFS and the most accurate approach, and GA-ML-CFS has the advantage of selecting substantially smaller feature subsets than the methods that obtained the most accurate result for each classifier.

7.2 Future Research Directions

Future research directions to extend our current work can be broadly divided into two groups: direct extensions of the ML-CFS methods proposed in this thesis (and the corresponding experiments), and new types of ML-CFS methods.

7.2.1 Direct Extensions of ML-CFS and GA-based ML-CFS

There are several direction extensions (or modifications) of ML-CFS that could potentially improve its performance. First, in all ML-CFS versions mentioned in this thesis, we used the arithmetic mean to measure an average value of correlation between a feature and all labels. The arithmetic mean is easy to implement and a widely used measure of a central tendency value, but it has the significant drawback of being very sensitive to outliers (extremely high or extremely low values) in the data set. In future work the median could be used instead of arithmetic mean, since the median is less sensitive to outliers.

Second, the ML-CFS methods that exploit biological knowledge were applied only on two microarray datasets. More experiments on other microarray datasets could be done in the future (if more multi-label microarray datasets become available). Moreover, other cancer-related databases could be used as a source of biological knowledge (different from the KEGG database used in Chapter 5), to try to improve the performance of ML-CFS.

Third, an extended versions of LexGA-ML-CFS with three objectives to be optimized could be implemented. This would be an extended version of the LexGA-MLCFS proposed in Chapter 6, by adding one more objective. Recall that in LexGA-ML-CFS, the first objective is the merit value and the second objective is the number of selected features. In the new three-objective approach, the merit value formula would be decomposed into three components corresponding to three objectives to be optimized in the following decreasing order of priority: the correlation between features and labels, the correlation between feature pairs, and the number of selected features.

Fourth, since the results reported in this thesis are limited by the use of two

specific multi-label classification algorithms (ML-kNN and BPMLL), in order to get broader computational results about the effectiveness of multi-label feature selection methods, other multi-label classification algorithms could be run in the future.

7.2.2 New Methods for ML-CFS

As mentioned in Section 6.1, hill-climbing search performs only a local search in the space of candidate feature subsets, selecting just one feature at a time and considering only one candidate solution at a time. On the other hand, Genetic Algorithms (GAs) are stochastic search methods which perform a more global search in the feature space than a greedy search. As a result of their global search, GAs cope better with feature interaction and are less likely to get trapped into a local optimum in the search space, being more likely to find a global optimum. However, GAs are not the only type of global search method, and other global search methods could be used to perform multi-label feature selection.

In particular, Ant Colony Optimization (ACO) is another type of global search and optimization method, which is inspired by the behaviour of a real ant colony in nature [28, 29, 30, 38, 81]. ACO algorithms also work with a population of candidate solutions (artificial ants) exploiting different regions of the search space. ACO has been extensively used for developing classification algorithms [48, 87, 88, 89, 98, 99]. In addition, there has been some work on ACO algorithms for conventional (single-label) feature selection [1, 3, 53, 60] and discovering multi-label classification rules [11, 12], but not yet for multi-label feature selection. Hence, it would be interesting to develop a new ACO algorithm for multi-label feature selection.

Moreover, another interesting research direction would be to develop new ML-CFS methods for hierarchical multi-label classification problems [102]. This type

of problem is more complex than conventional ("flat") multi-label classification problems, since in hierarchical multi-label classification the class labels are organized into a hierarchical structure – typically a tree or a directed acyclic graph of class labels. Hence, a new ML-CFS method for hierarchical multi-label classification would need to be able to cope with the hierarchical structure of class labels.

Bibliography

[1] Aghdam, M. H., Ghasem-Aghaee, N., and Basiri, M. E. Text feature selection using ant colony optimization. *Expert systems with applications 36*, 3 (2009), 6843–6853.

[2] Aksoy, S., and Haralick, R. M. Feature normalization and likelihood-based similarity measures for image retrieval. *Pattern Recognition Letters 22*, 5 (2001), 563–582.

[3] Al-Ani, A. Ant colony optimization for feature subset selection. In *WEC (2)* (2005), Citeseer, pp. 35–38.

[4] Babu, M. M. Introduction to microarray data analysis. *Computational Genomics: Theory and Application* (2004), 225–249.

[5] Bala, J., De Jong, K., Huang, J., Vafaie, H., and Wechsler, H. Using learning to facilitate the evolution of features for recognizing visual concepts. *Evolutionary Computation 4*, 3 (1996), 297–311.

[6] Bala, J., Huang, J., Vafaie, H., DeJong, K., and Wechsler, H. Hybrid learning using genetic algorithms and decision trees for pattern classification. In *IJCAI (1)* (1995), Citeseer, pp. 719–724.

[7] Bandyopadhyay, N., Kahveci, T., Goodison, S., Sun, Y., and Ranka, S. Pathway-based feature selection algorithm for cancer microarray data. *Advances in bioinformatics 2009* (2010).

[8] Berry, M. J., and Linoff, G. S. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 2004.

[9] Blickle, T. Tournament selection. *Evolutionary computation 1* (2000), 181–186.

[10] Boutell, M. R., Luo, J., Shen, X., and Brown, C. M. Learning multi-label scene classification. *Pattern Recognition 37*, 9 (2004), 1757–1771.

[11] Chan, A., and Freitas, A. A new classification-rule pruning procedure for an ant colony algorithm. In *Artificial Evolution* (2006), Springer, pp. 25–36.

[12] Chan, A., and Freitas, A. A. A new ant colony algorithm for multi-label classification with applications in bioinfomatics. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation* (2006), ACM, pp. 27–34.

[13] Chen, M.-S., Han, J., and Yu, P. S. Data mining: an overview from a database perspective. *Knowledge and data Engineering, IEEE Transactions on 8*, 6 (1996), 866–883.

[14] Cherkauer, K. J., and Shavlik, J. W. Growing simpler decision trees to facilitate knowledge discovery. In *KDD* (1996), vol. 96, Citeseer, pp. 315–318.

[15] Chuang, L.-Y., Yang, C.-H., Wu, K.-C., and Yang, C.-H. A hybrid feature selection method for dna microarray data. *Computers in biology and medicine 41*, 4 (2011), 228–237.

[16] Clare, A., and King, R. D. *Knowledge discovery in multi-label phenotype data*. Principles of data mining and knowledge discovery. Springer, 2001, pp. 42–53.

[17] Coenen, F. Data mining: past, present and future. *The Knowledge Engineering Review 26*, 01 (2011), 25–29.

[18] Correa Goncalves, E., Plastino, A., Freitas, A., et al. A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on* (2013), IEEE, pp. 469–476.

[19] Dash, M., and Liu, H. Feature selection for classification. *Intelligent data analysis 1*, 1 (1997), 131–156.

[20] Dash, M., and Liu, H. Consistency-based search in feature selection. *Artificial intelligence 151*, 1 (2003), 155–176.

[21] de Carvalho, A. C., and Freitas, A. A. *A tutorial on multi-label classification techniques*. Foundations of Computational Intelligence Volume 5. Springer, 2009, pp. 177–195.

[22] Dessì, N., and Pes, B. An evolutionary method for combining different feature selection criteria in microarray data classification. *Journal of Artificial Evolution and Applications 2009* (2009), 3.

[23] Dimou, A., Tsoumakas, G., Mezaris, V., Kompatsiaris, I., and Vlahavas, I. An empirical study of multi-label learning methods for video annotation. In *Content-Based Multimedia Indexing, 2009. CBMI'09. Seventh International Workshop on* (2009), IEEE, pp. 19–24.

[24] Dimou, A., Tsoumakas, G., Mezaris, V., Kompatsiaris, I., and Vlahavas, L. An empirical study of multi-label learning methods for video annotation. In *Content-Based Multimedia Indexing, 2009. CBMI'09. Seventh International Workshop on* (2009), IEEE, pp. 19–24.

[25] Ding, C., and Peng, H. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology 3*, 02 (2005), 185–205.

[26] Doquire, G., and Verleysen, M. *Feature selection for multi-label classification problems*. Advances in Computational Intelligence. Springer, 2011, pp. 9–16.

[27] Doquire, G., and Verleysen, M. Mutual information-based feature selection for multilabel classification. *Neurocomputing 122* (2013), 148–155.

[28] Dorigo, M., Birattari, M., and Stützle, T. Ant colony optimization. *Computational Intelligence Magazine, IEEE 1*, 4 (2006), 28–39.

[29] Dorigo, M., and Blum, C. Ant colony optimization theory: A survey. *Theoretical computer science 344*, 2 (2005), 243–278.

[30] Dorigo, M., and Stützle, T. Ant colony optimization: overview and recent advances. In *Handbook of metaheuristics*. Springer, 2010, pp. 227–263.

[31] Dziuda, D. M. *Data mining for genomics and proteomics: analysis of gene and protein expression data*, vol. 1. John Wiley Sons, 2010.

[32] Eiben, A. E., and Smith, J. E. *Introduction to evolutionary computing*. Springer Science & Business Media, 2003.

[33] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P. From data mining to knowledge discovery in databases. *AI magazine 17*, 3 (1996), 37.

[34] Freitas, A. A. *Data mining and knowledge discovery with evolutionary algorithms*. Springer Science Business Media, 2002.

[35] Freitas, A. A. *A survey of evolutionary algorithms for data mining and knowledge discovery*. Advances in evolutionary computing. Springer, 2003, pp. 819–845.

[36] Freitas, A. A. A critical review of multi-objective optimization in data mining: a position paper. *ACM SIGKDD Explorations Newsletter 6*, 2 (2004), 77–86.

[37] Freitas, A. A. *Evolutionary algorithms for data mining.* Data Mining and Knowledge Discovery Handbook. Springer, 2005, pp. 435–467.

[38] Freitas, A. A., Parpinelli, R. S., and Lopes, H. S. Ant colony algorithms for data classification. *Encyclopedia of Information Science and Technology 1* (2008), 154–159.

[39] George, G., and Raj, V. C. Review on feature selection techniques and the impact of svm for cancer classification using gene expression profile. *arXiv preprint arXiv:1109.1062* (2011).

[40] Gheyas, I. A., and Smith, L. S. Feature subset selection in large dimensionality domains. *Pattern recognition 43*, 1 (2010), 5–13.

[41] Glaab, E., Garibaldi, J. M., and Krasnogor, N. Learning pathway-based decision rules to classify microarray cancer samples.

[42] Grosan, C., and Abraham, A. *Intelligent Systems: A Modern Approach.* Intelligent Systems Reference Library. Springer Berlin Heidelberg, 2011.

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

[44] Hall, M. A. *Correlation-based feature selection for machine learning* (1999).

[45] Han, J., Kamber, M., and Pei, J. *Data mining: concepts and techniques: concepts and techniques.* Elsevier, 2011.

[46] Hand, D. J., Mannila, H., and Smyth, P. *Principles of data mining.* MIT press, 2001.

[47] Hernandez, J. C. H., Duval, B., and Hao, J.-K. A genetic embedded approach for gene selection and classification of microarray data. In *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics.* Springer, 2007, pp. 90–101.

[48] Holden, N., Freitas, A. A., et al. A hybrid particle swarm/ant colony algorithm for the classification of hierarchical biological data. In *Proceedings of the IEEE swarm intelligence symposium (SIS)* (2005), pp. 100–107.

[49] Hong, J.-H., and Cho, S.-B. The classification of cancer based on dna microarray data that uses diverse ensemble genetic programming. *Artificial intelligence in Medicine 36*, 1 (2006), 43–58.

[50] Hong, J.-H., and Cho, S.-B. Efficient huge-scale feature selection with speciated genetic algorithm. *Pattern Recognition Letters 27*, 2 (2006), 143–150.

[51] Huerta, E. B., Duval, B., and Hao, J.-K. A hybrid ga/svm approach for gene selection and classification of microarray data. In *Applications of Evolutionary Computing*. Springer, 2006, pp. 34–44.

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

[53] Jensen, R. Performing feature selection with aco. In *Swarm Intelligence in Data Mining*. Springer, 2006, pp. 45–73.

[54] John, G. H., Kohavi, R., Pfleger, K., et al. Irrelevant features and the subset selection problem. In *Machine Learning: Proceedings of the Eleventh International Conference* (1994), pp. 121–129.

[55] Jungjit, S., and Freitas, A. A lexicographic multi-objective genetic algorithm for multi-label correlation based feature selection. In *Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation* (New York, NY, USA, 2015), GECCO Companion '15, ACM, pp. 989–996.

[56] Jungjit, S., and Freitas, A. A. A new genetic algorithm for multi-label correlation-based feature selection. 285–290.

[57] Jungjit, S., Freitas, A. A., Michaelis, M., and Cinatl, J. A multi-label correlation-based feature selection method for the classification of neuroblastoma microarray data. In *Advances in Data Mining: 12th Industrial Conference (ICDM 2012) Workshop Proceedings Â? Workshop on Data Mining in Life Sciences (DMLS 2012).* (2012), IBAI Publishing, pp. 149–157.

[58] Jungjit, S., Michaelis, M., Freitas, A. A., and Cinatl, J. Two extensions to multi-label correlation-based feature selection: A case study in bioinformatics. In *Systems, Man, and Cybernetics (SMC), 2013 IEEE International Conference on* (2013), IEEE, pp. 1519–1524.

[59] Jungjit, S., Michaelis, M., Freitas, A. A., and Cinatl, J. Extending multi-label feature selection with kegg pathway information for microarray data analysis. In *Computational Intelligence in Bioinformatics and Computational Biology, 2014 IEEE Conference on* (2014), IEEE, pp. 1–8.

[60] Kanan, H. R., and Faez, K. An improved feature selection method based on ant colony optimization (aco) evaluated on face recognition system. *Applied Mathematics and Computation 205*, 2 (2008), 716–725.

[61] Kanehisa, M., and Goto, S. Kegg: kyoto encyclopedia of genes and genomes. *Nucleic acids research 28*, 1 (2000), 27–30.

[62] Kantardzic, M. *Data mining: concepts, models, methods, and algorithms.* John Wiley & Sons, 2011.

[63] Kim, S., Kon, M., DeLisi, C., et al. Pathway-based classification of cancer subtypes. *Biol Direct 7*, 1 (2012), 21.

[64] Kira, K., and Rendell, L. A. A practical approach to feature selection. In *Proceedings of the ninth international workshop on Machine learning* (1992), pp. 249–256.

[65] Kong, D., Ding, C., Huang, H., and Zhao, H. Multi-label relieff and f-statistic feature selections for image annotation. In *Computer Vision*

and Pattern Recognition (CVPR), 2012 IEEE Conference on (2012), IEEE, pp. 2352–2359.

[66] Kudo, M., and Sklansky, J. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition 33*, 1 (2000), 25–41.

[67] Langley, P., et al. *Selection of relevant features in machine learning*. Defense Technical Information Center, 1994.

[68] Lastra, G., Luaces, O., Quevedo, J. R., and Bahamonde, A. *Graphical feature selection for multilabel classification tasks*. Advances in Intelligent Data Analysis X. Springer, 2011, pp. 246–257.

[69] Lee, C.-P., and Leu, Y. A novel hybrid feature selection method for microarray data analysis. *Applied Soft Computing 11*, 1 (2011), 208–213.

[70] Lee, C.-P., Lin, W.-S., Chen, Y.-M., and Kuo, B.-J. Gene selection and sample classification on microarray data based on adaptive genetic algorithm/k-nearest neighbor method. *Expert Systems with Applications 38*, 5 (2011), 4661–4667.

[71] Lee, J., and Kim, D.-W. Feature selection for multi-label classification using multivariate mutual information. *Pattern Recognition Letters 34*, 3 (2013), 349–357.

[72] Li, D., Deogun, J. S., and Wang, K. Gene function classification using fuzzy k-nearest neighbor approach. In *Granular Computing, 2007. GRC 2007. IEEE International Conference on* (2007), IEEE, pp. 644–644.

[73] Li, L., Jiang, W., Li, X., Moser, K. L., Guo, Z., Du, L., Wang, Q., Topol, E. J., Wang, Q., and Rao, S. A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset. *Genomics 85*, 1 (2005), 16–23.

[74] Li, L., Liu, H., Ma, Z., Mo, Y., Duan, Z., Zhou, J., and Zhao, J. Multi-label feature selection via information gain. In *Advanced Data Mining and Applications*. Springer, 2014, pp. 345–355.

[75] Li, S., Wu, X., and Hu, X. Gene selection using genetic algorithm and support vectors machines. *Soft computing 12*, 7 (2008), 693–698.

[76] Liu, H., and Motoda, H. *Feature selection for knowledge discovery and data mining*. Springer Science Business Media, 1998.

[77] Liu, H., and Motoda, H. *Computational methods of feature selection*. CRC Press, 2007.

[78] Liu, H., Motoda, H., Setiono, R., and Zhao, Z. Feature selection: An ever evolving frontier in data mining. *FSDM 10* (2010), 4–13.

[79] Liu, H., and Yu, L. Toward integrating feature selection algorithms for classification and clustering. *Knowledge and Data Engineering, IEEE Transactions on 17*, 4 (2005), 491–502.

[80] Molla, M., Waddell, M., Page, D., and Shavlik, J. Using machine learning to design and interpret gene-expression microarrays. *AI Magazine 25*, 1 (2004), 23.

[81] Mullen, R. J., Monekosso, D., Barman, S., and Remagnino, P. A review of ant algorithms. *Expert Systems with Applications 36*, 6 (2009), 9608–9617.

[82] Ni, B., and Liu, J. A hybrid filter/wrapper gene selection method for microarray classification. In *Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on* (2004), vol. 4, IEEE, pp. 2537–2542.

[83] Ni, B., and Liu, J. A novel method of searching the microarray data for the best gene subsets by using a genetic algorithm. In *Parallel Problem Solving from Nature-PPSN VIII* (2004), Springer, pp. 1153–1162.

[84] Ogata, H., Goto, S., Fujibuchi, W., and Kanehisa, M. Computation with the kegg pathway database. *Biosystems 47*, 1 (1998), 119–128.

[85] Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., and Kanehisa, M. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic acids research 27*, 1 (1999), 29–34.

[86] Oh, I.-S., Lee, J.-S., and Moon, B.-R. Hybrid genetic algorithms for feature selection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 26*, 11 (2004), 1424–1437.

[87] Otero, F. E., Freitas, A. A., and Johnson, C. G. cant-miner: an ant colony classification algorithm to cope with continuous attributes. In *Ant colony optimization and swarm intelligence*. Springer, 2008, pp. 48–59.

[88] Parpinelli, R. S., Lopes, H. S., and Freitas, A. A. An ant colony based system for data mining: applications to medical data. In *Proceedings of the genetic and evolutionary computation conference (GECCO-2001)* (2001), Citeseer, pp. 791–797.

[89] Parpinelli, R. S., Lopes, H. S., and Freitas, A. A. An ant colony algorithm for classification rule discovery. *Data mining: A heuristic approach* (2002), 191–208.

[90] Peng, H., Long, F., and Ding, C. 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 (2005), 1226–1238.

[91] Pereira, R. B., Plastino, A., Zadrozny, B., and Merschmann, L. H. Information gain feature selection for multi-label classification. *Journal of Information and Data Management 6*, 1 (2015), 48.

[92] Prajapati, P., Thakkar, A., and Ganatra, A. A survey and current research challenges in multi-label classification methods. *International Journal of Soft Computing 2* (2012).

[93] Quinlan, J. R. *C4. 5: programs for machine learning*. Elsevier, 2014.

[94] Read, J. A pruned problem transformation method for multi-label classification. In *Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)* (2008), pp. 143–150.

[95] Read, J., Pfahringer, B., and Holmes, G. Multi-label classification using ensembles of pruned sets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on* (2008), IEEE, pp. 995–1000.

[96] Reza, F. *An Introduction to Information Theory*. Dover Books on Mathematics. Dover Publications, 2012.

[97] Saeys, Y., Inza, I., and Larranaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics (Oxford, England) 23*, 19 (Oct 1 2007), 2507–2517. LR: 20091104; JID: 9808944; RF: 135; 2007/08/24 [aheadofprint]; ppublish.

[98] Salama, K. M., and Freitas, A. A. Abc-miner: an ant-based bayesian classification algorithm. In *Swarm Intelligence*. Springer, 2012, pp. 13–24.

[99] Salama, K. M., and Freitas, A. A. Learning bayesian network classifiers using ant colony optimization. *Swarm Intelligence 7*, 2-3 (2013), 229–254.

[100] Shalabi, L. A., and Shaaban, Z. Normalization as a preprocessing engine for data mining and the approach of preference matrix. In *Depend-*

ability of Computer Systems, 2006. DepCos-RELCOMEX'06. International Conference on (2006), IEEE, pp. 207–214.

[101] Sharpe, P. K., and Glover, R. P. Efficient ga based techniques for classification. *Applied Intelligence 11*, 3 (1999), 277–284.

[102] Silla Jr, C. N., and Freitas, A. A. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery 22*, 1-2 (2011), 31–72.

[103] Spolaôr, N., Cherman, E. A., Monard, M. C., and Lee, H. D. Filter approach feature selection methods to support multi-label learning based on relieff and information gain. In *Advances in Artificial Intelligence-SBIA 2012*. Springer, 2012, pp. 72–81.

[104] Spolaor, N., Monard, M. C., Tsoumakas, G., and Lee, H. Label construction for multi-label feature selection. In *Intelligent Systems (BRACIS), 2014 Brazilian Conference on* (2014), IEEE, pp. 247–252.

[105] Spolaôr, N., Cherman, E. A., Monard, M. C., and Lee, H. D. A comparison of multi-label feature selection methods using the problem transformation approach. *Electronic Notes in Theoretical Computer Science 292* (2013), 135–151.

[106] Srinivas, M., and Patnaik, L. M. Adaptive probabilities of crossover and mutation in genetic algorithms. *Systems, Man and Cybernetics, IEEE Transactions on 24*, 4 (1994), 656–667.

[107] Srinivas, M., and Patnaik, L. M. Genetic algorithms: A survey. *Computer 27*, 6 (1994), 17–26.

[108] Tan, F., Fu, X., Zhang, Y., and Bourgeois, A. G. A genetic algorithm-based method for feature subset selection. *Soft Computing 12*, 2 (2008), 111–120.

[109] Tan, P.-N., Steinbach, M., and Kumar, V. *Introduction to data mining.* Pearson Addison Wesley, 2006.

[110] Tan, P.-N., Steinbach, M., and Kumar, V. *Introduction to data mining*, vol. 1. Pearson Addison Wesley Boston, 2006.

[111] Trohidis, K., Tsoumakas, G., Kalliris, G., and Vlahavas, I. P. Multi-label classification of music into emotions. In *ISMIR* (2008), vol. 8, pp. 325–330.

[112] Tsoumakas, G., and Katakis, I. Multi-label classification: An overview. *Dept.of Informatics, Aristotle University of Thessaloniki, Greece* (2006).

[113] Tsoumakas, G., Katakis, I., and Vlahavas, I. *Mining multi-label data.* Data mining and knowledge discovery handbook. Springer, 2010, pp. 667–685.

[114] Tsoumakas, G., Katakis, I., and Vlahavas, I. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering 23*, 7 (2011), 1079–1089.

[115] Wang, Y., Tetko, I. V., Hall, M. A., Frank, E., Facius, A., Mayer, K. F., and Mewes, H. W. Gene selection from microarray data for cancer classification—a machine learning approach. *Computational biology and chemistry 29*, 1 (2005), 37–46.

[116] Witten, I. H., and Frank, E. *Data Mining: Practical machine learning tools and techniques.* Morgan Kaufmann, 2005.

[117] Yang, C.-H., Chuang, L.-Y., and Yang, C. H. Ig-ga: a hybrid filter/wrapper method for feature selection of microarray data. *Journal of Medical and Biological Engineering 30*, 1 (2009), 23–28.

[118] Yang, C.-H., Chuang, L.-Y., Yang, C. H., et al. Ig-ga: a hybrid filter/wrapper method for feature selection of microarray data. *Journal of Medical and Biological Engineering 30*, 1 (2010), 23–28.

[119] Yu, L., and Liu, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML* (2003), vol. 3, pp. 856–863.

[120] Zhang, M.-L., Peña, J. M., and Robles, V. Feature selection for multi-label naive bayes classification. *Information Sciences 179*, 19 (2009), 3218–3229.

[121] Zhang, M.-L., and Wu, L. Lift: Multi-label learning with label-specific features.

[122] Zhang, M.-L., and Zhang, K. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining* (2010), ACM, pp. 999–1008.

[123] Zhang, M.-L., and Zhou, Z.-H. Multilabel neural networks with applications to functional genomics and text categorization. *Knowledge and Data Engineering, IEEE Transactions on 18*, 10 (2006), 1338–1351.

[124] Zhang, M.-L., and Zhou, Z.-H. Ml-knn: A lazy learning approach to multi-label learning. *Pattern Recognition 40*, 7 (2007), 2038–2048.

[125] Zhu, Z., Ong, Y.-S., and Dash, M. Markov blanket-embedded genetic algorithm for gene selection. *Pattern Recognition 40*, 11 (2007), 3236–3248.