Université de Tunis
Institut Supérieur de Gestion
Ecole Doctorale Sciences de Gestion

Stable and Efficient Feature Selection Methods for High Dimensional Data

THESE
En vue de l'obtention du Doctorat
En Informatique de Gestion

Présentée et soutenue publiquement par:

Afef BEN BRAHIM

Le 22 Juillet 2015

Membres du Jury:

♦*M. Mhamed-Ali El-Aroui*	*Professeur*	*à l'Université de Carthage*	*Président*
♦*M. Mohamed Limam*	*Professeur*	*à l'Université de Tunis*	*Directeur de thèse*
♦*M. Sami Faiz*	*Professeur*	*à l'Université de la Manouba*	*Rapporteur*
♦*M. Lamjed Ben Said*	*M. de Conf.*	*à l'Université de Tunis*	*Rapporteur*
♦*Mme Nadia Essoussi*	*M. de Conf.*	*à l'Université de Tunis*	*Membre*

Année Universitaire 2014 - 2015

Thèse / ISGT / Juillet 2015
Stable and Efficient Feature Selection Methods For High Dimensional Data
Afef BEN BRAHIM

Acknowledgements

I am thankful to Allah, most Gracious, who in His infinite mercy has guided me to achieve this PhD work.

I dedicate this thesis to my loving, supportive, encouraging, and patient parents, Jamila and Mohamed Habib. I would like to tell them thank you for all their love and sacrifices. They raised me with a love of science and supported me in all my pursuits. This thesis would not be possible without their presence and support.

I also deeply thank my supervisor Pr. Mohamed Limam. It has been an honor to be his Ph.D. student. I could not be prouder of my academic roots and hope that I can in turn pass on the research values he has given to me. He has been very patient in supervising this work. I appreciate all his contributions of time and ideas to make my Ph.D. experience productive and stimulating.

I am also deeply grateful to my beloved sisters, Imen, Inès, Wafa and Yosra. They supported me during my thesis with their love and encouragements. I also dedicate this thesis to my beloved nephews Nada, Youssef, Mehdi, Nour and Ons.

I would also like to give my special thanks to all my family and friends for their encouragements, especially to Zeineb Chelly, Randa Regaya, Waad Bouaguel and Abir Smiti.

I also thank Pr. Alexandros Kalousis for his work on stable feature selection and for all his valuable comments on my work.

Also, thanks to all LARODEC members.

I finally dedicate this thesis to all those who supported me, encouraged me and believed in me.

Table of Contents

List of Figures

List of Tables

List of Algorithms

List of Acronyms

AROM	Approximation of the zeRO-norm Minimization
CAA	Classification Accuracy based Aggregation
CBFS	Correlation Based Feature Selection
CFS	Candidate Feature Subset
CLA	Complete Linear Aggregation
CNS	Central Nervous System
CSS	Cooperative Subset Search
CV	Cross Validation
DLBCL	Diffuse Large Bcells
ECA	Ensemble Classifier Aggregation
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
GAA	Genetic Algorithm based Aggregation
HIB-CSS	Hybrid Instance Based CSS
HIB-SBS	Hybrid Instance Based SBS
IB-Filter	Instance Based Filter
IG	Information Gain
kNN	k-Nearest Neighbors
MCE	Miss-Classification Error
mRMR	minimum Redundancy Maximum Relevance

NH	Nearest Hit
NM	Nearest Miss
NPV	Negative Predictive Value
OFA	Occurrence Frequency based Aggregation
OOB	Out Of Bag
PCA	Principal Component Analysis
PK-mRMR	Prior-Knowledge based mRMR
PK-Relief	Prior-Knowledge based Relief
PK-RFE	Prior-Knowledge based RFE
PPV	Positive Predictive Value
PS-l2-AROM	Partially Supervised l2 AROM
RAA	Reliability Assessment based Aggregation
RFE	Recursive Feature Elimination
ROC	Receiver Operating Characteristic
RRA	Robust RankAggregate
SBS	Sequential Backward Search
SFFS	Sequential Forward Floating Search
SFS	Selected Feature Subset
SS-l2-AROM	Semi Supervised l2 AROM
SVM	Support Vector Machines
SVM.RFE	SVM and RFE
TN	True Negative
TP	True Positive
WMA	Weighted Mean Aggregation
WMV	Weighted Majority Vote
WRA	Weight-Rank Aggregation

Introduction

The rapid technological developments in different life domains increased the amounts of data at an unprecedented speed. This may appear useful for the decision making process, however it is not the case when this increase concerns dimensions of data. Examples of such data are measurements arising in character, text, face recognition from digitized images, spam email identification, diagnostic tasks in medicine and genetic engineering, recognition tasks in biology, economics, astronomy, etc. In microarray data analysis for example, each sample involves the measurements of tens of thousands of variables corresponding to the expression of tens of thousands of genes measurable with microarray technology. This results in high dimensional data sets with small sample size. Unfortunately, existing machine learning methods are not designed to handle such data setting, because the ability to build models with scientific validity is negatively impacted by an increasing ratio between the number of variables and the sample size. This phenomenon, known as the curse of dimensionality, is based on the fact that high dimensional data is often difficult to work with. A large number of features can increase the noise of the data and thus the error of a learning algorithm, especially if there are only few observations compared to the number of features.

Feature selection is a solution for such problems where there is a need to reduce the number of features and thus the dimensionality of the data. Feature selection reduces data dimensionality by removing irrelevant and redundant features. It aims at improving algorithms predictive accuracy, and increasing the constructed models comprehensibility.

A great variety of feature selection algorithms have been developed with a focus on improving the predictive accuracy of learning models while reducing dimensionality and

model complexity. However, most of existing methods do not take into account the small sample size problem in their design. Yet, this data specificity produces some problems, not only for predictive performance of learning algorithms, but also results in the instability of feature selection results. Stability of feature selection is its insensitivity to variations in the training set. This issue is particularly critical for application domains like microarray data analysis, where feature selection is used as a knowledge discovery tool for identifying robust biomarkers. For this reason, besides the predictive accuracy, researchers are increasingly drawing attention to stability of feature selection.

A Summary of Major Contributions

This thesis focuses on the design of methods achieving stable feature selection while allowing estimation of models with good classification performance for high dimensional small sample size data. These methods propose several means to handle the lack of enough samples in that high-dimensional setting.

The first contribution is based on instance learning. We propose three approaches, one filter and two hybrid algorithms. Their main challenge is to convert the problem of small sample size to a tool that allows choosing few subsets of features to be combined or analyzed in order to select the most relevant ones. Each instance proposes a candidate subset of the most relevant features where small sample size makes this process feasible with acceptable running time. Thus, the high dimensionality of data is reduced to few subsets of features such that their number corresponds to the data sample size.

The second contribution is based on ensemble methods. We proceed by a comparative study between different aggregation levels of ensemble feature selection, classifier and selector levels. Our objective is to study the effect of the aggregation level on the classification performance. Then, we focus on ensemble selector aggregation level by proposing a robust feature aggregation method to combine the results of different feature subsets. This approach takes advantage of multiple classifier system benefits to enhance the classification accuracy. First, an ensemble of different feature subsets are obtained by a function or data perturbation. After this step, a multiple classifier system is trained on each of the projections of the resulting feature subsets on the training data. An evaluation protocol is used to obtain the classification performance of each setting. This classification performance

is used to measure the reliability of selected features. Initial feature weights are adjusted based on the features' corresponding reliability and a final subset is obtained by selecting the best features from different individual subsets based on their adjusted weights.

In our third contribution and in order to obtain robust feature selection results, we propose to incorporate prior knowledge about some dimensions known to be more relevant, as a means of directing the feature selection process. We propose prior knowledge based extensions of three well known feature selection techniques. We propose also a robust embedded feature selection method based on prior knowledge. This method makes use of a partial supervision on some features. Prior knowledge about these dimensions known to be more relevant is incorporated as a means of guiding the feature selection process. Iteratively we make use of the initial prior knowledge and the previously selected features to expand a subset of highly relevant features in a pre-processing phase of feature selection.

Thesis Overview

This thesis is made of four chapters. Chapter 1 gives some machine learning background. Chapter 2 dicusses instance based feature selection. Chapter 3 is dedicated to the ensemble feature selection and the new proposed robust aggregation technique and Chapter 4 aims at enhancing feature selection stability by incorporating prior knowledge.

Chapter 1

Background and related work

Contents

1.1 Introduction

This chapter discusses the main aspects of feature selection. Section 1.2 introduces the basics of feature selection and states some differences between feature selection versus extraction and univariate versus multivariate selection. Three widely used selection patterns: filters, wrappers and embedded methods are presented and their representative methods are described. Section 1.3 details several feature selection evaluation metrics for assessing feature selection quality in the context of classification. Classification performance and its metrics and stability definition, causes, existing studies and metrics are discussed. Section 1.4 presents different validation protocols along with the situations where each can be used and the justification for the chosen protocol. Section 1.5 concludes the chapter.

1.2 Data mining and feature selection

Data mining refers to the process of analyzing data from different aspects and summarizing it into useful information. This is done by the research of correlations or patterns among the fields in large relational databases. Data mining consists of three major steps which are preprocessing, mining, and post-processing. Feature selection is frequently used as a preprocessing step to data mining. It is the process of reducing the whole feature space to a subset of relevant features based on some evaluation measures.

Dimensionality reduction can be done by either feature selection or feature extraction (Guyon and Elisseff (2003), van der Maaten et al. (2008)). Feature extraction reduces data dimensionality by projecting the data into lower dimensional space formed by combinations of features. It is a successful technique to reduce dimensionality and improve learning performance. However, the new feature space is not physically linked to the original features. Hence, there is a problem of interpretability and the familiar meaning of features is lost. Feature selection, on the other hand selects a subset of the original features without any kind of transformation (Guyon and Elisseff (2003), Saeys et al. (2007)). Therefore, the selected features keep their original meaning and interpretation.

Researchers have been interested in developing feature selection methods since 1970's as this process has shown effectiveness in eliminating irrelevant and redundant features, increasing efficiency of learning process, improving learning performance like predictive

accuracy and enhancing comprehensibility of learned results (Kohavi and John (1997), Blum and Langley (1997) and Dash and Liu (1997)).

Generally, the feature selection process consists of four basic steps, namely subset generation, subset evaluation, stopping criterion, and result validation (Dash and Liu (1997) Langley (1994)). Subset generation is based on a search procedure which generates candidate feature subsets (CFS). The search step is the most time consuming in the feature selection process. The subset evaluation guides the choice of relevant features based on some quality criterion. Stopping criteria can be based on a generation procedure. For example, algorithm stops if a predefined number of features are selected or a predefined number of iterations is reached. It can also be based on the evaluation function. For example, algorithm stops if an optimal subset is obtained according to some evaluation function or if addition or deletion of any feature does not produce a better subset. The choice of a suitable quality criterion is important to optimize the feature selection process. Then, the selected subset is validated using a test set from synthetic or real world data. Figure (1.1) summarizes the four steps of the feature selection process as described by Dash and Liu (1997).

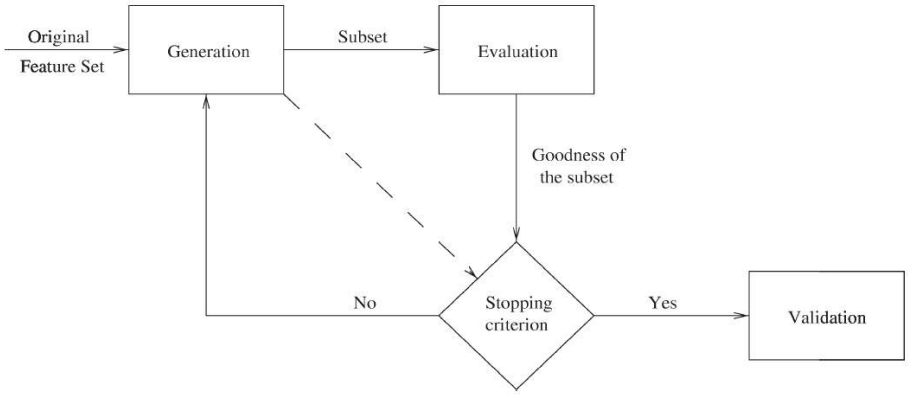


Figure 1.1: The feature selection process.

Feature selection is used with many data mining functions such as classification, clustering, association rules and regression. It is also a well-studied research area in statistics where it is called variable selection.

Feature selection is specially useful with high dimensional data where there are thousands of features (Kohavi and John (1997)). In such data sets, it is necessary to find an

optimal feature subset. Many feature selection algorithms have been proposed in the literature and have proved their efficiency in improving the performance of learning models built on the selected features (Saeys et al. (2007), Guyon and Elisseff (2003)).

Selection vs. Extraction: Feature extraction methods can also be used to reduce dimensionality. Feature extraction, as defined by Wyse et al. (1980), "consists of the extraction a set of new features from the original features through some functional mapping". Principal component analysis (PCA) is a well known feature extraction method (Jolliffe (1986)). It uses an orthogonal transformation to convert a set of observations of possibly correlated features into a set of values of linearly uncorrelated features called principal components. A learning algorithm is then applied using the resulting dimensions. PCA and most of feature extraction techniques use unsupervised learning. They do not take into account the target class and thus do not aim to improve the classification performance. Moreover, these techniques reduce the dimensionality with a loss of interpretability as the resulting dimensions are obtained by a certain combination of the original features (Dash and Liu (1997)). For these reasons, feature extraction is not considered and we are concerned with supervised learning where one of the main objectives of feature selection is to improve classification performance.

Univariate vs. Multivariate: Feature selection methods can be categorized to univariate or multivariate (Guyon and Elisseff (2003)). Univariate methods evaluate relevancy of each feature relevance independently of others while multivariate methods take into account features dependencies while evaluating them. If d is the total number of features, the complexity of univariate methods is generally $O(d)$, while multivariate approaches are more complex specially for data sets with large d as they introduce all feature dependencies. In that sense, univariate features are advantageous. However, in several cases, taking interactions between features into consideration is very important to select features. Guyon and Elisseff (2003) provide several examples where univariate methods miss such interactions between features. They concluded that two features apparently useless on their own can be useful with others. In biological data, features are genes that influence each other. Thus, multivariate feature selection methods are more suitable to such applications.

There are three supervised feature selection categories namely, wrappers, filters and embedded methods. Many reviews of these methods are found in the literature. Guyon

and Elisseff (2003), Saeys et al. (2007) and Kohavi and John (1997) are examples of such good reviews. Filters select subsets of features as a pre-processing step, independently of the chosen predictor. Wrapper and embedded methods, on the other hand, generally use a specific learning algorithm to evaluate a specific subset of features. The following subsections briefly introduce these different categories.

1.2.1 Filters

Filters are the simplest of the three approaches described in this chapter (Kohavi and John (1997), Guyon and Elisseff (2003)). For filter methods, measuring the relevance of a feature subset is not time consuming. Filters are not dependent of a specific type of predictive model. They only take characteristics of the data into consideration to select a best feature subset or to obtain a feature's ranking by assigning a score to each feature. This is done before the learning process begins. Filter methods are very fast and thus very useful to select features in high dimensional data sets. Their principle is illustrated in Figure (1.2).

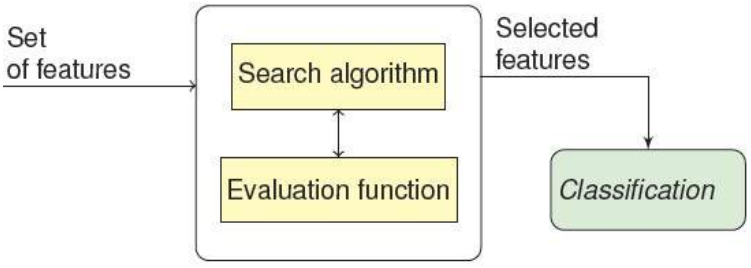


Figure 1.2: Feature selection: The Filter Model.

A filter algorithm first ranks features based on some quality criteria. Features with the highest weights or ranks are then selected to induce classification. Feature evaluation could be either univariate or multivariate. As discussed above, in the univariate scheme each feature is evaluated independently of the others, while the multivariate scheme evaluates features in batches. Therefore, the multivariate scheme is naturally capable of handling feature redundancy. Several filter methods have been proposed in the literature and have shown their effectiveness on selecting the most relevant features and improving the predictive performance. Some of the most popular filter methods are described in the following.

Fisher score: (Duda et al. (2001)) This filter selects features, such that distances between instances in different classes are as large as possible, while distances between instances in the same class are as small as possible. With this definition, the score for the j^{th} feature Sc_j is calculated by Fisher score as follows:

$$Sc_j = \frac{\sum_{k=1}^{K} n_k(\mu_{jk} - \mu_j)^2}{\sum_{k=1}^{K} n_k \rho_{jk}^2} \tag{1.1}$$

where μ_{jk} and ρ_{jk} are the mean and the variance of the j^{th} feature in the k^{th} class respectively. The n_k is the number of instances in the k^{th} class, and μ_j is the mean of the j^{th} feature. The algorithm selects the top ranked features based on the obtained scores.

t-test filter: Another filter technique commonly used is the statistical t-test (Gosset (1908)). It is traditionally used to compare two normally distributed samples or populations. It prefers features with a maximal difference of mean value between groups and a minimal variability within each group. The t-test is used in the form that defines the score of a feature as the ratio of the difference between its mean values for each of the two classes and the standard deviation. The latter takes into account the standard deviation values of the feature for every class and its cardinality. The weight of each feature is thus given by its computed absolute score.

Information gain: Information gain (IG) is one of the most popular feature selection methods based on mutual information (Quinlan (1993)). IG is simple and computationally efficient. It measures the information between the j^{th} feature f_j and the class labels C, i.e. the amount of information in bits about the class prediction, in the presence of that feature and knowing the corresponding class distribution. Given S the set of training examples, IG of a feature is calculated as follows:

$$IG(S, f_j) = H(S) \sum_{v=values(f_j)} \frac{\mid S_{f_j=v} \mid}{\mid S \mid} H(S_{f_j=v}), \tag{1.2}$$

where $\frac{|S_{f_j=v}|}{|S|}$ is the fraction of examples with f_j having the value v, and $H(S)$ is the entropy given by:

$$H(S) = -\sum_{k=1}^{K} p(c_k) log_2(p(c_k)), \tag{1.3}$$

where $p(c_k)$ is the probability of observing class c_k in the training set S and K is the number of classes. $H(S_{f_j=v})$ is calculated the same way using only the subset of instances with f_j having the value v. A feature is relevant if it has a high IG. Features are selected in a univariate way, therefore IG cannot handle redundant features.

Minimum-Redundancy-Maximum-Relevance: The minimum-Redundancy-Maximum-Relevance (mRmR) method proposed by Peng et al. (2005) is also a mutual information based method. It selects features according to the maximal statistical dependency criterion. The mRMR method selects a feature subset that has the highest relevance with the target class, subject to the constraint that selected features are mutually as dissimilar to each other as possible. Given f_j, representing the attribute j, and the class label C, their mutual information is defined in terms of their frequencies of appearances $p(f_j)$, $p(C)$, and $p(f_j, C)$ as follows

$$I(f_j, C) = \int p(f_j, C) \log \frac{p(f_j, \omega)}{p(f_j)p(C)} df_j dC. \tag{1.4}$$

Maximum-Relevance method selects the best individual features correlated to the class labels by finding a feature set S with n features, which jointly has the largest dependency $D(S, C)$, on the target class C given by:

$$\max D(S, C), D = \frac{1}{|S|} \sum_{f_j \in S} I(f_j, C). \tag{1.5}$$

However, those top features may have high correlations with each other. In order to remove the redundancy among features, a Minimum-Redundancy criterion, $minR(S)$, is introduced where mutual information between each pair of attributes is taken into consideration. This criterion is given by

$$\min R(S), R = \frac{1}{|S|^2} \sum_{f_j, f_t \in S} I(f_j, f_t). \tag{1.6}$$

Assume that A represents the whole feature set and we have already selected S_{n-1}, the feature set with $n-1$ features. In order to choose the n^{th} feature from the set $\{A - S_{n-1}\}$, the two constraints D and R are combined and the feature maximizing this combination is selected as follows

$$\max_{f_j \in A - S_{n-1}} [I(f_j, C) - \frac{1}{n-1} \sum_{f_j \in S_{n-1}} I(f_j, f_t)]. \tag{1.7}$$

An incremental process is used to select features satisfying optimization criteria of Eqs. (1.6) and (1.7). The m^{th} feature can also be selected as follows:

$$\max_{f_j \in A - S_{n-1}} [I(f_j, C) / \frac{1}{n-1} \sum_{f_j \in S_{n-1}} I(f_i, f_t)]. \tag{1.8}$$

By combining optimization criteria of Eqs. (1.6) and (1.7), mRMR improves the generalization properties of the features in the subset and the classification performance.

Correlation-Based Feature Selection: The Correlation Based Feature Selection (CBFS) method searches for subsets of features that are individually highly correlated with the class but have low inter-correlation (Hall (2000)). Thus, like mRMR, it is a multivariate filter which takes into account interactions between features in order to eliminate redundancy. Correlation coefficients are used to estimate correlations between subsets of attributes and classes as well as inter-correlations between the features. Iteratively, CBFS selects features that have the highest correlation with the class based on some measure such as conditional entropy. The relevance of a group of features grows with the correlation between features and classes and decreases when inter-correlation becomes high. CBFS is usually combined with search strategies such as forward selection, backward elimination, bi-directional search, best-first search and genetic search. The following equation :

$$Merit_{S_n} = \frac{n\overline{r_{cf}}}{\sqrt{n + n(n-1)\overline{r_{ff}}}}, \tag{1.9}$$

gives the merit of a feature subset S consisting of n features where $\overline{r_{cf}}$ is the average value of all feature-classification correlations, and $\overline{r_{ff}}$ is the average value of all feature-feature correlations. The CBFS criterion selects the best subset as follows:

$$CBFS = \max_{S_n} \left[\frac{r_{cf_1} + r_{cf_2} + \cdots + r_{cf_n}}{\sqrt{n + 2(r_{f_1 f_2} + \cdots + r_{f_j f_t} + \cdots + r_{f_n f_1})}} \right], \tag{1.10}$$

where r_{cf_j} and $r_{f_j f_t}$ variables are referred to as correlations, and could be Pearson's correlation coefficient or Spearman's ρ.

Instance based feature selection : Relief: This method was proposed by Kira and Rendell (1992). Relief and its multi-class extension ReliefF (Kononenko (1994)) are based on instance learning. They select features to separate instances from different classes. Assume that m instances are randomly sampled from the data, the score S_j of the j^{th} feature is defined by Relief as:

$$S_j = \frac{1}{2}\sum_{i=1}^{m} d(X_{ji} - X_{jNM_i}) - d(X_{ji} - X_{jNH_i}), \tag{1.11}$$

where NM_i denotes the values on the j^{th} feature of the nearest instances to the sample X_i with different class labels, while NH_i denotes the values on the j^{th} feature of the nearest instances to X_i with the same class label and $d(.)$ is a distance measure. ReliefF handles multi-class problem by extending Eq. (1.11) as follows,

$$S_j = \frac{1}{K}\sum_{i=1}^{m}(-\frac{1}{m_i}\sum_{x_t \in NM_i} d(X_{ji} - X_{jt}) + \sum_{c \neq c_i}\frac{1}{h_{ic}}\frac{p(c)}{1-p(c)}\sum_{x_t \in NH_{ic}} d(X_{ji} - X_{jt})) \tag{1.12}$$

where K is the number of classes, NM_i and NH_{ic} denote the sets of nearest points to X_i with different classes c and with the same class, with sizes m_i and h_{ic} respectively, and $p(c)$ is the probability of an instance belonging to the class c. Robnik and Kononenko (2003) related the relevance evaluation criterion of Relief to the hypothesis of margin maximization, which explains why the algorithm provides superior performance in many applications.

It is argued that filters, compared to wrappers, are faster and that some filters provide a generic selection of features not tuned for a given learning algorithm. Another advantage of filters is that they can be used as a preprocessing step to reduce space dimensionality and overcome overtting (Kohavi and John (1997), Guyon and Elisseff (2003)). When the number of features becomes very large, the filter model is usually chosen as it is computationally efficient, fast and independent of the classification algorithm. The crucial issue when using filters is the choice of a criterion function. Given that each of the filters uses a specific feature evaluation criterion, we may not say that a resulting subset is better than the others but rather that all the obtained subsets are the best subsets among the whole feature space. Also, taking into account the predictive performance of a learning algorithm while selecting features could be of a big interest, since enhancing this performance is one of

the main objectives of feature selection. Filters ignore this aspect and this is their major shortcoming.

1.2.2 Wrappers

It is of high interest that the search for the optimal feature subset takes into account the specific biases and performance of the predictive algorithm. Based on this, wrapper models use a specific classifier to evaluate the quality of selected features (Kohavi and John (1997)). The performance measure of a learning algorithm along with a statistical resampling technique such as cross validation (CV) (Kohavi (1995)) are used to select the best feature subset. Given a predefined classifier, a typical wrapper model performs the following steps:

- 1. Producing a set of features based on a searching procedure,
- 2. Evaluating features using the performance of a classifier,
- 3. Repeating Step 1 and Step 2 until a feature set with the desired quality is reached,
- 4. Evaluating the final feature set using the classifier on an independent testing set.

A general framework for wrapper methods of feature selection for classification is shown in Figure (1.3).

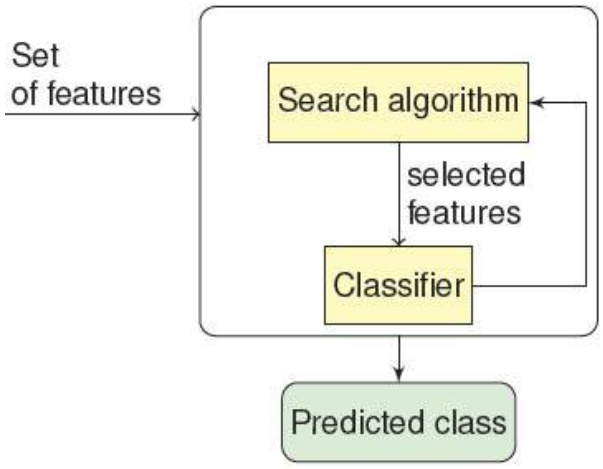


Figure 1.3: Feature selection: The Wrapper Model.

In wrapper models, the predictive classifier works as a black box, its performance with the selected features will be returned back to the feature search component for the next iteration of feature subset selection. The complexity of the search procedure for d features is $O(2^d)$, thus an exhaustive search is impractical unless m is small (Guyon and Elisseff (2003)). A wide range of search strategies can be used and are described in the following.

Sequential feature selection: It is one of the most widely used wrapper techniques (Kohavi and John (1997), Blum and Langley (1997), Aha and L. (1996)). It selects a subset of features by forward or backward search, which consist on respectively adding or removing features until certain stopping conditions are satisfied. In the sequential forward selection process, single attributes are added to an initially empty set of attributes. The sequential backward elimination works in the opposite direction of forward selection. Starting from the full set, the feature that results in the smallest decrease in the value of the objective function is sequentially removed.

Randomized Hill-Climbing: Compared to sequential wrapper methods, randomized wrapper algorithms search the next feature subset at random (Skalak (1994)). Single features or several features can be added at once, removed, or replaced from the previous feature set based on the effect on the predictive performance. With these updates, the current set moves to the subset with the highest accuracy. The search procedure terminates when no subset improves over the current set.

Genetic Algorithm: Genetic Algorithm (GA) selects features by optimizing the prediction error of the model built upon the set of selected features (Vafaie and Jong (1992)). GA is inspired by the natural evolution, it models a dynamic population of solutions Holland (1975). The three basic operators of GA are: selection, crossover and mutation. The members of the population, referred to as chromosomes present the selected features. The error of the model built using each chromosome serves as a fitness function. In the evolution phase, the chromosomes are subjected to crossover and mutation. The algorithm exchanges and recombines a pair of chromosomes through crossover. The mutation is an operator which allows diversity. It alters one or more feature values in a chromosome from its initial state. The probability for a chromosome to mutate should be predefined and should not be high, otherwise the search will turn into a primitive random search. The algorithm minimizes the error function in resulting generations. Several factors are necessary

to the success of GA feature selection process. The choice of initial population is important as well as the choice of parameters guiding the different algorithm steps. A careful choice of these parameters allows the population to explore the solution space and to avoid early convergence to a local minimum. Also, we need to choose suitable crossover and mutation probability time.

Wrappers usually provide the best performing feature set for a particular type of model and have the ability to take into account feature dependencies as they consider groups of features jointly. However, the lack of generality of wrappers is a drawback. Different learning algorithms could lead to different feature selection results. Additionally, wrappers repeatedly build learning models on each CFS. Thus, they are time consuming and this is their major problem, especially if building the learning algorithm has a high computational cost as reported by Saeys et al. (2007). Moreover, many wrapper methods require the training of a large number of classifiers and manual specification of many parameters. This makes their implementation and use rather complicated requiring expertise in machine learning (Kohavi and John (1997)). This is probably one of the main reasons why filter methods are more popular in many domains such as bioinformatics.

1.2.3 Embedded methods

The embedded approach (Guyon and Elisseff (2003) Lal et al. (2006)), as it is the case for wrappers, selects features according to a learning algorithm. However, while the search and the evaluation procedures are separated in the wrapper model, the search for an optimal subset of features is built into the classifier construction using its internal parameters. For this reason, they are less computationally intensive than wrappers. During the learning process, each feature is assigned a weight indicating its relevance. At the end of this process, the set of most relevant features are returned as optimal feature set. The embedded model is described in Figure (1.4).

Decision trees: Decision trees are classification algorithms that use embedded feature selection. A decision tree is iteratively built by splitting the data depending on the value of a specific feature chosen according to its discriminative power of separating different classes. A widely used criterion for the importance of a feature is the mutual information between feature f_j and the outputs Y. This procedure is repeated recursively on the feature

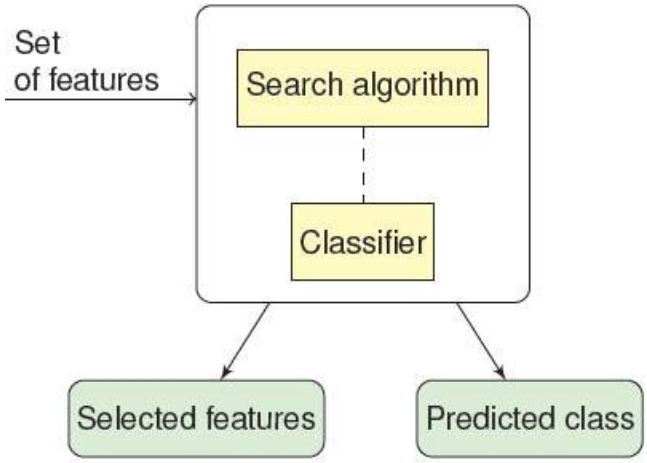


Figure 1.4: Feature selection: The Embedded Model.

subsets until some stopping criterion is satisfied. The output is a model that uses only a subset of features that appear in the nodes of the decision tree. Decision trees are thus not sensitive to the presence of irrelevant features and feature selection is implicitly built into the algorithm. Therefore, decision trees can be considered as an embedded method. The most popular approaches include CART introduced by Breiman and Stone (1984), ID3 (Quinlan (1986)) and C4.5 (Quinlan (1986)).

Support vector machines and recursive feature elimination: Guyon et al. (2002) introduced an embedded feature selection using the weight vectors of a Support Vector Machine (SVM) (Vapnik (1995)) in combination with recursive feature elimination (RFE) to form SVM.RFE where the ranking criterion is computed for all features based on their corresponding weights. This process is iterated and the features with the smallest rankings, i.e. weights, are removed. The remaining features are selected. This iterative procedure is a backward feature elimination (Kohavi and John (1997)). The algorithm can be accelerated by removing more than one feature. Guyon et al. (2002) applied SVM.RFE to the task of gene selection, and results have shows that their method eliminates gene redundancy automatically and yields subsets that achieve better classification than the full set of features.

Regularization models: There are also embedded models based on regularization (Lal et al. (2006)), also called penalization methods that introduce additional constraints into the optimization of a predictive algorithm. Simultaneously, the objective function minimizes

the corresponding errors and estimates a coefficient vector w with properly tuned penalties. Each coefficient w_j corresponds to one feature f_j and some coefficients can be exactly equal to zero. Features with coefficients that are close to zero are then eliminated. Feature selection is achieved this way and only features with nonzero coefficients in w will be used in the final classier. Examples of regularization algorithms are the Approximation of the zeRO-norm Minimization (AROM) methods (Weston et al. (2003), Helleputte and Dupont (2009b)) and LASSO method (Lal et al. (2006)).

The AROM methods: Given m examples $x_i \in R^m$ and the corresponding class labels $y_i \in \{-1, +1\}$ with $i = 1, ..., m$, a linear model $g(x)$ predicts the class of any point $x_i \in R^m$ as follows:

$$g(x) = sign(w \cdot x_i + b) \tag{1.13}$$

where b is the point where the line crosses the $g(x)$ axis. Feature selection is closely related to a specific form of regularization of this decision function to enforce sparsity of the weight vector w. Weston et al. (2003) studied in particular the zero-norm minimization subject to linear margin constraints:

$$min_w \parallel w \parallel_0 subjectto : y_i(w \cdot x_i + b) \geq 1 \tag{1.14}$$

where $\parallel w \parallel_0 = card\{\omega_j \mid \omega_j \neq 0\}$ and card is the set cardinality. Since Problem (1.14) is NP-Hard, a log $l1$-norm minimization is proposed instead as follows

$$min_w \sum_{j=1}^{n} \ln(\mid \omega_j \mid +\epsilon) subjectto : y_i(w.x_i + b) \geq 1 \tag{1.15}$$

where $0 < \epsilon \leq 1$ is added to smooth the objective when some $\mid \omega_j \mid$ vanishes. The natural logarithm in the objective function facilitates parameter estimation with a simple gradient descent procedure. The resulting algorithm $l1$-AROM iteratively optimizes the $l1$-norm of w with rescaled inputs.

The $l2$-AROM method further approximates this optimization by replacing the $l1$-norm by the $l2$-norm. A smooth feature selection occurs during an iterative process where the weight coefficients along some dimensions progressively drop below the machine precision while other dimensions become more significant. A final ranking of the absolute values of each dimension can be used to obtain a fixed number of features (Weston et al. (2003)).

Lasso regularization: Lasso regularization (Lal et al. (2006)) is based on $l1$-norm of the coefficient of w and defined as:

$$penalty(w) = \sum_{j=1}^{d} \mid w_j \mid \tag{1.16}$$

An important property of the $l1$ regularization is that it can generate an estimation of w with exact zero coefficients. In other words, there are zero entities in w which denotes that the corresponding features are eliminated during the classier learning process. Therefore, it can be used for feature selection.

Embedded methods have the advantage of including the interaction with the classification model and thus selecting features that improve the predictive performance. Additionally, they are usually less computationally expensive than wrapper methods. However, the computational complexity remains a major problem when the number of features becomes very high. Other issues including algorithm implementation also remain.

1.2.4 Hybrid methods

One major issue with wrapper methods is their high computational complexity due to the need to train a large number of classifiers. With a very high number of features, a hybrid approach could be adopted that follows filter model in the search step selecting a small number of CFS. Then, a wrapper method is applied to the candidate subsets to achieve the best possible performance with a particular learning algorithm leading to a less complex model. Accordingly, the hybrid model is more efficient than filter and less expensive than wrapper. A combination of a filter criteria and a classifier may produce a new hybrid technique.

Xie et al. (2010) proposed the improved F-score and Sequential Forward Floating Search (SFFS) which combines F-score with SFFS and SVM to select relevant features, where the improved F-score is an evaluation criterion for filters, while SFFS and SVM constitute an evaluation system of wrappers for the diagnosis of erythemato-squamous diseases.

Huang et al. (2007) proposed a hybrid GA with two stages of optimization where the mutual information between the predictive labels and the true classes serves as the fitness

function for the GA in the wrapper step and an improved estimation of the conditional mutual information acts in a filter manner.

1.3 Performance metrics

This section describes several feature selection quality criteria, namely interpretation, classification accuracy and stability. Feature selection results are interpretable in many applications and selection methods leading to meaningful results should be preferred to those that do not. However, in some cases the interpretability of the selected features requires a deep knowledge of the application field that a computer scientist may not have, making this evaluation not evident. For example, in the case of genomic data, the interpretation of selected genes requires first, the availability of a corresponding number of studies and annotations and second, a deep knowledge of the molecular biology domain. This makes the interpretation of a selected subset of genes not easy for a computer scientist. Additionally, interpretation is difficult to apply to a method, because it generally takes time to interpret a single or many subsets of selected features. For these reasons, we choose to use two other metrics, which are classification accuracy and stability, as our main evaluation criteria for feature selection. We are interested in methods' evaluation more than interpretability of results. However, the classification accuracy evaluation relies essentially on a predictive algorithm. Thus, we define the predictive models used in this thesis then we describe several metrics for assessing classification performances and stability in the next subsections.

1.3.1 Classification algorithms

Data classification is categorised under supervised learning where the objective is the prediction of a class membership value, also called class label of unknown observations or samples using a training data for which the class labels are known. Each observation in the training or test data is represented by a feature vector associated with it. Different classification algorithms use different techniques for finding relations between the predictor features' values and the class labels in the training data. These relations are summarized in a model which will be applied to new cases to predict their class labels. Thus, the process of data classification involves generally two steps: training of a classifier and testing of the trained classifier. There are many different classifiers that can be applied in different

applications. Two of the most efficient classification algorithms in data mining (Wu et al. (2007)) and high dimensional small sample size data sets are discussed in the following.

k-nearest neighbor classification: k-nearest neighbors (kNN) classifier developed by Cover and Hart (1967) and originally proposed by Fix and Hodges (1951), is a simple learning procedure which searches for a group of k objects in the training set that are nearest to the test sample, and assigns a class label based on the most represented class in this neighborhood. The value of k, the number of nearest neighbors, is fixed by the user. The search for k neighbours is based on a distance or similarity metric to compute distance between objects. The choice of a distance measure depends on the type of data.

Let DS be a training set of size m where each instance is given by $\mathbf{x}_i = (x_{i1}, \ldots, x_{id}) \in \mathbb{R}^d$, where d is the number of features, and $\mathbf{y}_i = (y_1, \ldots, y_m)$ the vector of class labels for the m instances. Given a test object $z = (z_j, \ldots, z_d)$, the algorithm computes the distance between z and all the training objects x_i to determine its k nearest-neighbors. The most used distance metric for data with discrete and continuous features is the Euclidean distance which measures distance between objects as follows:

$$dist(z, x_i) = \sum_{j=1}^{d} \sqrt{(z_j - x_{ij})^2}. \tag{1.17}$$

Other distance measures can be used depending on the available data. Once the k nearest-neighbors are obtained, the test object is assigned a class $y\prime$ based on classes of its nearest neighbors. The majority vote is the most popular rule used to search for predominant class in the neighborhood. The majority vote rule is applied as follows:

$$y\prime = argmax_{\omega} \sum_{(x_i, y_i) \in DS} f(\omega = y_i) \tag{1.18}$$

where ω is a class label, y_i is the class label for the i^{th} nearest neighbor and $f(.)$ is a function that returns the value 1 if its argument is true and 0 otherwise.

SVM: This classification method, introduced by Vapnik (1995), is one of the most robust machine learning techniques. There are many reasons for using SVM classifier. It requires less prior assumptions about the input data and can perform on small or huge data set by doing a nonlinear mapping from an original input space into a high dimensional feature space. An SVM model is a representation of the examples as points in space, mapped

so that examples of separate classes are divided by a clear gap that is as wide as possible found by maximizing the margin between the two classes. Finding the maximum margin hyperplanes offers the best generalization ability. New examples will then be mapped into that same space and predicted to belong to a class based on which side of the gap they fall in. Thus, it consists of a linear classification function which corresponds to a separating hyperplane $f(x)$ that passes through the middle of the two classes, separating the two. More formally, a new data instance x_i is classified by simply testing the sign of the function $f(x_i)$, x_i belongs to the positive class if $f(x_i) > 0$.

The search for the maximum margin hyperplanes is done by maximizing the following function with respect to w and b:

$$Lp = \frac{1}{2} \parallel w \parallel - \sum_{i=1}^{m} \alpha_i y_i (w.x_i + b) + \sum_{i=1}^{m} \alpha_i. \tag{1.19}$$

where m is the number of training examples, and $\alpha_i, (i = 1, ..., t)$ are non-negative numbers such that the derivatives of Lp with respect to α_i are zero, α_i are Lagrange multipliers and Lp is called the Lagrangian. In this equation, the vectors w and constant b define the hyperplane.

1.3.2 Classification performance

Classification performance is an important evaluation criterion of feature selection. Several classification performance metrics can be found in the review of Costa et al. (2007). Let us assume that a set of possible class labels consists of positive, p, and negative, n, labels. The total number of positives is P and the total number of negatives is N. For example, in a disease diagnosis classification task, an instance is assigned to a positive class, in case of the diagnosis of a disease (e.g., cancer), while it is assigned to the negative class for the case of no disease, i.e. healthy. There are four possible outcomes of a classification algorithm for this problem:

- True positive (TP) : If the instance is positive and it is classified as positive.
- False negative (FN): If the instance is positive and it is classified as negative.
- False positive (FP): If the instance is negative and it is classified as positive.

- True negative (TN): If the instance is negative and it is classified as negative.

This representation is given by comparing true and predicted labels. This comparison is done by using a confusion matrix where lines and columns are respectively true and predicted classes. TP and TN represent correct decisions made by the classifier, while FN and FP are classification errors. From this matrix, several useful characteristics of classification performance can be derived which include:

Accuracy : The most used performance criterion is the correct classification rate, known as accuracy. This measure ranges from 0, with perfect misclassification, to 1 when the classifier perfectly classifies testing data. Its popularity is certainly due to its simplicity, not only in terms of processing but also of interpretation, since it corresponds to the observed proportion of correctly classified instances. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{P + N} \tag{1.20}$$

Sensitivity and specificity : Sensitivity or recall is the proportion of TP over the total number of positive cases. In the case of the disease prediction, it is an index of the performance of a diagnostic test, calculated as the percentage of individuals with a disease who are correctly classified as having the disease. Specificity is the proportion of TN over the total number of negative cases. For the same example, it is calculated as the percentage of individuals without the disease, who are classified as not having the disease. These two measures are given by:

$$Sensitivity = recall = \frac{TP}{P} \tag{1.21}$$

$$Specificity = 1 - \frac{FP}{N} = \frac{TN}{TN + FP} \tag{1.22}$$

Positive Predictive Value (PPV): or precision is the probability that a person having a positive result on a diagnostic test actually has a particular disease. It is given by:

$$PPV = \frac{TP}{TP + FP} \tag{1.23}$$

Negative Predictive Value (NPV): It is the probability that a person having a negative result on a diagnostic test does not have the disease. It is given by:

$$NPV = \frac{TN}{TN + FN} \tag{1.24}$$

F-measure : The F-measure can be interpreted as a weighted average of the precision (PPV) and recall (sensitivity). It reaches its best value at 1 and worst score at 0. It is defined as:

$$F_{measure} = \frac{2}{1/precision + 1/recall} \tag{1.25}$$

Area under the Receiver Operator Characteristic curve : Another frequently used characteristic of a classifier, is the Receiver Operating Characteristic (ROC) curve. It is a characteristic allowing to visualize classification performance of one or several algorithms. A ROC curve is a plot of the sensitivity (or the TP rate) against one minus its specificity (or the FP rate). This plot depicts relative trade-offs between TPs and FPs.

1.3.3 Stability

Classical methods of classification break down when the dimensionality is extremely large. Fan and Fan (2008) show that for the independence classification rule, models using all features can be as poor as random guessing due to noise accumulation in high-dimensional feature space. They prove that almost all linear discriminants can perform as poorly as random guessing. Thus, it is important to select a subset of important features for high-dimensional classification. Besides, the curse of dimensionality phenomenon has a negative impact on stability of feature selection which is defined as the sensitivity of a method to variations in the training set (Kalousis et al. (2007)). Thus, stability helps studying the robustness of a selection method by measuring the similarity between sets of selected features obtained with variations such as sampling variations, or algorithm's parameters variations. A high stability means that the considered sets are highly similar, and vice-versa.

In biomarker discovery, stability of feature selection is an important selection quality metric for several reasons, among which the reproducibility and the easiness of the biological validation of the discovered biomarkers. The underlying hypothesis is that small changes in the data sampling should not result in considerable changes in the set of selected markers. This is rational since patients, belonging to a given class, are supposed to have a metabolism that is regulated the same way. Also, high stability is needed for example in

order to design a unique and small diagnosis kit. Therefore, the confidence in discovered markers is poor if there is instability of selection results (He and Yu (2010)).

Stability also helps studying how similar several selection methods are. If two selection methods tend to produce similar feature sets in many conditions, their behaviors can be thought as being close.

1.3.3.1 Causes of instability

He and Yu (2010) have highlighted three different aspects related to stability in feature selection:

- Ignoring stability aspect when designing feature selection algorithm: Enhancing the predictive accuracy of a classifier is the main objective for classical feature selection methods. Ignoring stability in the algorithm design results in instable feature selection.
- Existence of multiple subsets of good features: Different but highly correlated features may be selected in multiple subsets under different settings. Also, multiple feature subsets with relevant and non-correlated features can exist.
- High dimensional data with few samples: This problem is also called the curse of dimensionality and it has been proved that it is one of the main sources of instability in feature selection results.

Another source of feature selection instability is the existence of different selection criteria that can be used by a feature selection algorithm process. Two algorithms based on different selection relevance criteria may select different subsets of features for a same training set, each one optimizing the corresponding relevance criterion.

1.3.3.2 Existing studies on feature selection stability

During the last decade, many methods for stable feature selection have been proposed. A review of this methods is given by He and Yu (2010). Among them, methods that are based on ensemble learning and methods that incorporate prior feature relevance into the algorithm design stage. The group feature selection is another approach which handles data

with highly correlated features. In order to address the small data size problem, the sample injection method is also proposed. These four approaches are described in the following.

Ensemble feature selection: Dietterich (2000) defined ensemble learning as a machine learning method that imitate human's second nature to consult several persons before making a final decision in different life domains such as medicine, finance, social problems, etc. In machine learning, ensemble learning methods combine multiple learned models under the same assumption. Breiman (1996) proposed Bagging and Freund and Schapire (1997) introduced boosting which are two popular ensemble methods for classification. Ensemble feature selection techniques use the same concept by combining the selections of several feature selectors, often obtained by varying the selection algorithm or the training data.

Feature selection by incorporating prior knowledge: In many classification areas, experts may have prior knowledge about the relevance of some features. This constitutes a means to guide feature selection even that available knowledge concerns only a fraction of the features. Traditional feature selection algorithms tend to ignore prior knowledge about features. It has been shown that the use of prior knowledge on relevant features induces a large gain in stability with improved classification performance (Helleputte and Dupont (2009b)). Transfer learning can also be used to obtain such prior knowledge from different but related data sets (Helleputte and Dupont (2009a)).

Group feature selection: This approach identifies groups of correlated features in high dimensional data, and then performs feature selection by treating each consensus feature group as a single entity. These entities resist to variations of training samples. This leads to more stable feature selection (Loscalzo et al. (2009)).

Feature selection with sample injection: We have cited earlier that curse of dimensionality is one of the main sources of instability in feature selection. Thus, having more samples will naturally increase stability. However, in many applications like biomarker discovery, the generation of real sample data from patients and healthy people is usually expensive and time-consuming. With these constraints, researchers thought naturally of using other alternatives to achieve the same objective. He and Yu (2010) described two data augmentation strategies are possible. One is using test data to increase the sample size in feature selection process. Another method is to generate some artificial training samples according to the distribution of available training data.

1.3.3.3 Stability measure

For stable feature selection, one important issue is how to measure the stability of feature selection algorithms, i.e., how to quantify the selection sensitivity to variations in the training set.

Measuring stability requires a similarity metric that will measure to which extent K sets of selected features share common features. This measure depends on the representation language used by a given feature selection algorithm to describe its selection output (Kalousis et al. (2007)). Different selection output forms call for different similarity measures. There are three types of feature selection outputs, a feature weighting, a feature ranking obtained by sorting feature weights in a descending order, or simply a subset of features which by setting a threshold on the ranks or predefining a given subset size. Let training examples be described by a vector of features $S = (f_1, f_2, ..., f_d)$ where d is the total number of features, then a feature selection algorithm produces either:

- a weighting-scoring: $w = (w_1, w_2, .., w_d)$,
- a ranking: $r = (r_1, r_2, .., r_d), 1 \leq r_j \leq d$,
- or a subset of features: $s = (s_1, s_2, .., s_d), s_j \in 0, 1$, with 0 indicating absence of a feature and 1 presence.

In order to measure stability we need a measure of similarity for each of the above representations. To measure similarity between two weightings $w, w\prime$, produced by a given feature selection algorithm, we can use Pearson's correlation coefficient (Bonett and Wright (2000)):

$$Stab_W(w, w\prime) = \frac{\sum_j (w_j - \mu_w)(w\prime_j - \mu_{w\prime})}{\sqrt{\sum_j (w_j - \mu_w)^2 \sum_j (w\prime_j - \mu_{w\prime})^2}}, \tag{1.26}$$

where $Stab_W$ takes values in $[-1, 1]$; a value of 1 means that the weightings are positively correlated, a value of 0 that there is no linear correlation while a value of -1 that they are negatively correlated.

To measure similarity between two rankings $r, r\prime$, Spearman's rank correlation coefficient can be used (Spearman (1987)), and it is given by:

$$Stab_R(r, r\prime) = 1 - 6 \sum_j \frac{(r_j - r\prime_j)^2}{d(d^2 - 1)}, \tag{1.27}$$

where r_j and $r\prime_j$ are the ranks of feature f_j in rankings r and $r\prime$ respectively. The possible range of values is $[-1, 1]$. A value of 1 means that the two rankings are identical, a value of 0 that there is no correlation between the two ranks, and a value of -1 that they have exactly inverse orders.

Finally similarity between two subsets of features, that are produced by selecting features from different samples of the data, can be measured using Kuncheva stability index proposed by Kuncheva (2007) and defined as:

$$Stab_S(S_k, S_u) = \frac{\mid S_k \cap S_u \mid) - \frac{s^2}{d}}{s - \frac{s^2}{d}}, \tag{1.28}$$

where d is the total number of features, and S_k, S_u are two feature sets built from different partitions of the training samples. The ratio $\frac{s^2}{d}$ corrects the bias of selecting common features in both sets by chance. This correction motivates our use of this particular stability index. This index satisfies $-1 < Stab \leq 1$ and the greater is its value the larger is the number of commonly selected features in the two sets. A negative stability index means that feature sets sharing common features are mostly due to chance. This index can be generalized to K signatures as follows:

$$Stab_S(S_1, .., S_K) = \frac{2}{K(K-1)} \sum_{k=1}^{K-1} \sum_{u=k+1}^{K} \frac{\mid S_k \cap S_u \mid) - \frac{s^2}{d}}{s - \frac{s^2}{d}}. \tag{1.29}$$

The most interesting stability estimation is provided by $Stab_S$ since it focuses on a subset of features, the ones selected by each method, which is of interest when performing feature selection. It is also more practical to use $Stab_S$ when the objective is also to compare stability of several feature selection method. Nevertheless this estimation is specific to a given number of selected features. To get a more global picture of the stability profile of different methods with respect to $Stab_S$ we can compute its values for different sizes of selected feature sets. In this thesis, we focus on measuring $Stab_S$ to measure stability of feature selection and to compare our proposed methods to existing ones.

1.4 Validation settings

The evaluation of feature selection methods should be based on some evaluation metrics like those we discussed above but also on a set of validation data. The evaluation criteria will be quantified on the projection of the selected features on this validation data. If data used for validation is the same or belong to the training set on which feature selection has been applied, this will result in optimistically biased performance estimates and overfitting. Thus, the approach to be avoided is to estimate a model and its performance on the same data. Two possibilities can be considered instead: the holdout setting and the CV setting.

1.4.1 Holdout setting

Holdout, also called test sample estimation, splits the whole data set into a training and a test set also called holdout set. This is appropriate when the the size of available data is quite large. In this case, feature selection and classifier training are done on the training set, then the performance of the obtained model is evaluated on the test samples. However, this setting is not suitable with few samples available, where splitting the data into training and testing set will result on a weaker model than if it had been built on the whole data set and its performance will be poorly evaluated as the testing set is also very small. This leads to a weak model with a very limited generality. When the the number of available samples is small, incorporating the whole data in a validation loop is preferable. There exist some procedures to do this without biasing the model performance.

1.4.2 Cross validation

In the K-Fold CV setting (Stone (1974) Kohavi (1995)), a sample is split into K non-overlapping subsets of points and K models are built. Each model is built on all subsets except the k^{th} one. Performances of model k are evaluated on the k^{th} subset, unused for that model estimation. Note that if we perform feature selection on all the data and then cross validate, then the test data in each fold of the CV procedure is also used to choose the features and this is what biases the performance analysis. However, if we adopt the proper procedure, and perform feature selection in each fold, there is no longer any information about the held out cases in the choice of features used in that fold. The K independently

measured performances are then averaged. This resulting average is supposed to be equal to the performance of a model built on the available training data and tested on an infinite test set. The variability of the estimated performance can be computed via the standard deviation of the K performances, since the K test sets are independent and they do not overlap. To summarize, the key idea is that CV is a way of estimating the generalisation performance of a process for building a model, so we need to repeat the whole process in each fold. Otherwise we will end up with a biased estimate.

For a computer scientist, this setting is ideal to globally compare several methods, in terms of classification accuracy but also in terms of stability measured on the feature subsets obtained in multiple algorithm runs. It is however not practical for the biologist which looks for a unique feature set and a single predictive model. However, Kalousis et al. (2004) showed that with this validation protocol, there will be as many feature selection outputs and predictive models as validation loops. In this case, it is best to view the loop validation setting as assessing the performance of a procedure for fitting a model rather than the model itself. The best trade-off would be to perform CV in order to estimate the expected performance of the model building process, and then build the final model using the entire data set using the same procedure used in each fold of the CV. This is meaningful if the feature selection is stable. However, if the feature set varies greatly from one fold of the CV to another, it is an indication that the feature selection is unstable.

1.4.3 Bootstrap

This resampling technique has been also proposed to overcome the small sample size validation issue (Kohavi (1995)). This procedure draws a series of what is called bootstrap samples. Each bootstrap sample corresponds to the drawing of m observations with repetitions. To each bootstrap sample corresponds a set of points not drawn. This set forms what is called an out-of-bag sample (OOB). A model is trained on the bootstrap sample and evaluated on the OOB sample. This operation is repeated B times, and the average out-of-bag error is obtained.

1.4.4 Which setting to use?

Validation protocols are used in order to evaluate the efficiency of the different models proposed in this thesis. The current applicative context in which we investigate feature selection concerns high dimensional small sample size problems. As discussed above, the training-test setting is not appropriate in such problems as it leads to a weak model with a very limited generality. Thus, it is more appropriate to use a resampling technique like bootstrapping or CV. We opt for the K-fold CV setting with $K = 10$ which is adequate to our type of data, as detailed above. The key point here is to get an unbiased performance estimate. The procedure to use to generate the final model must be repeated in its entirety independently in each fold of the CV procedure. Also using this protocol will result on multiple sets of features each corresponding to a fold. If the feature set varies greatly from one fold of the CV to another, it is an indication that the feature selection is unstable and probably not very meaningful. Thus, we use CV also as a protocol for assessing feature selection stability.

1.5 Conclusion

In this chapter, the main aspects of feature selection are reviewed. We described the different categories of feature selection and the representative methods of each one. Then, we discussed the main evaluation criteria of feature selection and the available validation protocols giving the advantages and shortcomings of each and the justification for our choices. The next chapter proposes stable and efficient feature selection based on instance learning.

Chapter 2

Instance Based Feature Selection

Contents

2.1 Introduction

This chapter discusses instance based methods for feature selection on high dimensional data with few samples. Existing methods are not specially conceived to handle the small sample size of the data which is also one of the main causes of feature selection instability. In order to deal with the data small size problem, we propose three approaches, one filter and two hybrid algorithms. Their main challenge is that they convert the problem of the small sample size to a tool that allows choosing only a few subsets of features to be combined or analyzed in order to select the most relevant ones. Each instance proposes a candidate subset of the most relevant features for this instance. Small sample size makes this process feasible with acceptable running time. Thus, the high dimensionality of data is reduced to few subsets of features, where their number corresponds to the data sample size and this is when small sample size is of benefit to feature selection process.

2.2 Cancer classification and the small sample size problem

In cancer classification, a predictive model is built on the training data consisting of patients belonging to healthy or cancerous categories. The classification algorithm build the model by finding the relationship between the features which are gene expression profiles and the two class labels. Based on the learnt relationships, the model built will be able to separate the healthy and the cancerous class labels to diagnose cancer (Okun (2011)). As there are thousands of gene expressions and only few samples in a typical gene expression data set, serious problems occur with the application of many traditional statistical methods. Overfitting on the classifier is one of these problems. It leads to very good and often perfect classification performance on the training data, but this perfect performance does not translate to new unlabeled data resulting in a very limited classifier generalization.

Kohane et al. (2003) explained that the small sample size in genomic applications may be due to the high cost of the microarrays. Each sample involves the measurements of tens of thousands of variables corresponding to the expression of tens of thousands of genes measurable with microarray technology. The result is a large number of features compared to the number of samples. Kohane et al. (2003) described this system as highly underdetermined system and explained that based on the relatively small number of observations,

there is a large number of solutions in which the genes being measured could interact. Thus, due to the underdetermined nature of these systems, standard machine learning techniques do not hold up well because those techniques were developed under the assumption that the number of samples, m, is much larger than the features dimensionality d.

Since it is difficult to increase m for the reasons explained above, dimension reduction is a solution for such problem. It reduces the possibility for noisy and irrelevant genes to be included into one of the many possible solutions. Reducing the number of genes will reduce the algorithms variance. So, machine learning and more specially feature selection methods are useful to deal with high dimensional data sets.

2.2.1 Examples of gene expression data sets

Here, some of popular gene expression data sets are briefly described in order to give a realistic picture of what gene expression data are. These data sets will be used later for our experiments. The classification in these data sets is binary and its task is cancer diagnosis.

- Diffuse large Bcells (DLBCL): In this data set presented by Shipp et al. (2002), the classification task is the prediction of the tissue types, where genes are used to discriminate DLBCL tissues from Follicular Lymphomas.

- Bladder cancer dataset : The task in the Bladder cancer data set described by Dyrskjot et al. (2003) is the clinical classification of bladder tumors using microarrays.

- Lymphoma data set : Its task is to discriminate between two types of Lymphoma based on gene expression measured by microarray technology as in Alizadeh et al. (2000). This dataset contains missing values for numeric attributes that we replace using the kNN imputation method proposed by Troyanskaya et al. (2001). This method takes advantage of the correlation structure in the data and uses the average of records that have similar completed data patterns to impute missing values. The kNN imputation method is quite simple but is effective and often preferred over traditional and some of the more sophisticated methods. It has the advantage of assuming that data are missing at random, missing data only depends on the observed data, which in turn means that it is able to take advantage of multivariate relationships in the completed data.

- Prostate cancer data set: This data set described by Singh et al. (2002) contains expression level of 12600 genes for 102 samples including prostate tumors and normal samples.

- Breast cancer data set: This data set is used by van 't Veer et al. (2002) and its classification task is the diagnostic of the presence of Breast cancer disease. The dimensionality in this data set is about 24482 features which characterize 97 samples.

- The Central Nervous System (CNS): This data set is described in Pomeroy et al. (2002). It is a large data set concerned with the prediction of central nervous system embryonal tumor outcome based on gene expression.

- The malignant pleural mesothelioma and lung adenocarcinoma gene expression data base (Lung cancer): It is used by Gordon et al. (2002) and its task is to differentiate between malignant pleural mesothelioma and lung adenocarcinomas.

Table (2.1) summarizes the characteristics of the seven data sets.

Table 2.1: Datasets characteristics

Dataset	DLBCL	Bladder	Lymphoma	Prostate	Breast	CNS	Lung
No. of samples	77	31	45	102	97	60	181
No. of features	7029	3036	4026	12600	24482	7129	12533

2.2.2 Facing small sample size problem while selecting features

Learning in the small sample case is of practical interest. One reason for this is the difficulty in collecting data for each object. It's true that there are problems with such applications, however, can we find any "advantage" on working with small sample size data when performing feature selection? The answer is yes. There is a feature selection concept that can take advantage of the small sample size of data which is example or instance based feature selection. The key idea is to decompose an arbitrarily complex problem into a set of locally ones through local learning of feature relevance, and then find relevant features globally.

We propose three approaches, one filter and two hybrid algorithms based on this concept. Their main challenge is that they convert the problem of the small sample size to a tool that allows choosing only a few subsets of features to be combined or analyzed in order to select the most relevant ones. Each instance proposes a candidate subset of the most relevant features for this instance. Small sample size makes this process feasible with acceptable running time. Thus the high dimensionality of data is reduced to few subsets of features which number corresponds to the data sample size and this is when small sample size is of benefit to feature selection process.

2.3 Filter solutions based on instance learning

Among feature selection methods, filters rank all variables in terms of relevance, as measured by a score which depends on the method. They are simple to implement and fast to run. To obtain a signature of size n, one simply takes the top genes according to the score. Instance based filters are considered as one of the most effective feature selection algorithms (Li and Lu (2009), Hu et al. (2012)). A well known algorithm that relies on relevance evaluation is Relief (Kira and Rendell (1992)). Relief algorithm, based on random selection, assigns a relevance weight to each feature to denote the relevance of the feature to the target concept. For each feature, it samples instances randomly from the training set and updates the relevance values based on the difference between the selected instance and the two nearest instances of the same and opposite class. Then, the feature is scored as the sum of weighted differences in the different class and the same class. It has been proven that Relief is an online algorithm for a convex optimization problem (Sun et al. (2010)). By maximizing the averaged margin of the nearest patterns in the feature scaled space, Relief can estimate the feature weights in a straightforward and efficient manner. However, Relief based methods suffer from instability as the feature selection is based on instances that are picked at random. The feature weight may fluctuate with the instances (Robnik and Kononenko (2003)), making the selection sensitive to the data sampling especially in the presence of noisy and high-dimensional outliers. Moreover, Relief does not help with removing redundant features. As long as features are deemed relevant to the class concept, they will all be selected even though many of them are highly correlated to each other.

To take advantage of the strength of Relief algorithm on finding high relevant features

while overcoming the problem of stability and redundancy, we propose filter approaches that rely on Relief's feature weighting technique as a way of scoring features according to each instance. Using this weighting process, each instance will propose a feature ranking and the proposed algorithm will then focus on top ranked features for each instance. Lists of ranked features will finally be analyzed or combined to give a final feature subset. Three different combination schemes are proposed for combining selected feature subsets. They are based on features' occurrence frequency calculation and weights or ranks combination. A fourth proposed approach is the use of mutual information in a second step to eliminate redundancy among the selected features. The proposed approaches aim at obtaining a feature selection that yields good classification performance while being stable. These methods are based on a first common step which is candidate subsets construction. This step along with the different proposed combination schemes are described in the next sections.

2.3.1 Candidate subsets construction by instance based feature weighting

In a preprocessing step of the optimal feature subset selection, the feature space is reduced to few candidate subsets using instance based (IB) feature weighting. Each instance of the training data is an expert which proposes a candidate feature subset (CFS) based on an instance feature weighting technique . Let X be a matrix containing m training instances $\mathbf{x}_i = (x_{i1}, \ldots, x_{id}) \in \mathbb{R}^d$, where d is the number of features, and $\mathbf{y}_i = (y_1, \ldots, y_m), i = 1, ..., m$ the vector of class labels for the m instances. Let F be the set of features $\mathbf{f}_j = (f_1, \ldots, f_d), j = 1, ..., d$, where $d >> m$.

Given a distance function, we find the two nearest neighbors of each sample x_i for each feature f_j, one from the same class (called nearest hit or NH), and the other from the different class (called nearest miss or NM). Using a distance function, the margin of x_{ij} is then computed as:

$$W(x_{ij}) = d(x_{ij}, NM(x_{ij})) - d(x_{ij}, NH(x_{ij})). \tag{2.1}$$

For simplicity, we use the Manhattan distance to define a sample's margin and nearest neighbors. This weight definition is used in the Relief algorithm (Kira and Rendell (1992)) using the Euclidean distance, and it has been argued that there is not any significant difference noticed in the estimations use of Relief algorithms using the two metrics (Robnik and

Kononenko (2003)). These scores are then normalized and we obtain a weighted feature space for each instance x_i.

This weight is then projected on each feature f_j and we get the matrix $\mathbf{W}$ filled with feature weights $w_{j,i}$ as shown in Table 2.2. Once the algorithm completes the weighting

Table 2.2: Matrix of feature weights

	$\mathbf{f}_1$	$\mathbf{f}_2$	...	$\mathbf{f}_d$
$\mathbf{x}_1$	$w_{1,1}$	$w_{2,1}$	...	$w_{d,1}$
$\mathbf{x}_2$	$w_{1,2}$	$w_{2,2}$	...	$w_{d,2}$
$\mathbf{x}_3$	$w_{1,3}$	$w_{2,3}$	...	$w_{d,3}$
...	...	...	...	...
$\mathbf{x}_m$	$w_{1,m}$	$w_{2,m}$	...	$w_{d,m}$

process, features in the space of each instance are ranked based on their weights such as top ranked features are those with highest relevance weight. Note that a feature f_j may have different weights and ranks depending on the instance considered. Based on this IB feature weighting step, a candidate subset of cardinality n is chosen from the best ranked features of each instance. This pre-processing step leads to m candidate feature subsets $\{CFS_1, CFS_2, ...CFS_m\}$ of cardinality n. These candidate subsets must be analyzed or combined in some manner in order to obtain a final result. The proposed combination techniques are detailed in the following section.

2.3.2 Combination of candidates feature subsets

In this step, four different alternatives are proposed to obtain the final feature subset. Three of them are combination schemes that aggregate the candidate feature subsets. The last one is a redundancy detection filter, based on mutual information, applied to candidate subsets to get the final feature selection result.

2.3.2.1 Feature selection by calculating feature occurrence frequency

In this combination scheme based on feature occurrence frequency (OF), the m candidate subsets component features are first gathered together into a single subset $S = (f_1, f_2, ..., f_s)$ which is the union of all candidates. The resulting subset is projected on the m instances

such as for an instance x_i, the feature $f_k \in S$ is assigned 1 if it is selected in CFS_i and 0 otherwise. The final feature selection is obtained by calculating the number of occurrences of each feature over all instances and ranking them based on their occurrence frequency. This ranking technique favors features appearing in the maximum number of candidate feature subsets built based on instances. Thus, if new instances are tested, the selected features will most likely be relevant for classifying them. As the ranking technique is based on aggregating several opinions (candidate subsets), as explained earlier, our approach will improve feature selection stability. The final feature selection step is illustrated in Figure 2.1. We refer to this approach as IB-OF.

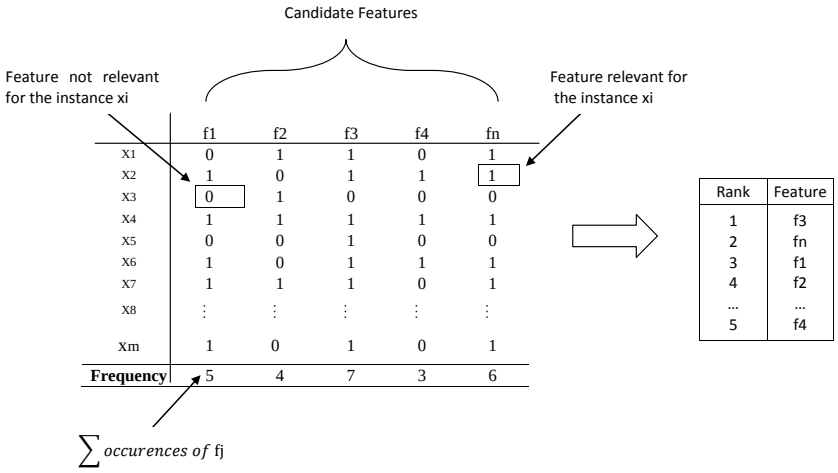


Figure 2.1: Feature selection based on calculating feature occurrence frequency

2.3.2.2 Feature selection by weighted mean aggregation

The weighted mean aggregation (WMA) method uses the weights of features obtained in the IB weighting step. First, the subset S is projected on the m instances such as for an instance x_i, the feature $f_k \in S$ is assigned the weight $(w_{k,i} = w_{j,i})$ if it corresponds to a feature f_j selected in CFS_i and 0 otherwise. Then, for each feature f_k the weights mean is calculated as follows:

$$WM(f_k) = \frac{\sum_{i=1}^{m}(w_{k,i})}{m}. \tag{2.2}$$

Features with the highest weighted mean are selected as final result. We refer to the resulting filter as IB-WMA.

2.3.2.3 Feature selection by ranks aggregation

The rank aggregation (RA) method uses the complete ranking of the features in S. Each CFS_i obtained after the IB weighting step consists of the n best ranked features for the instance x_i. Let $R_i = (r_i^1, r_i^2, ..., r_i^n)$ be the list providing a ranking of features in CFS_i. The best ranked feature is assigned rank 1, and the worst one rank n.

As for the aggregation methods described above, we consider the subset S and project it on the m instances. Then, for each instance x_i, the feature $f_k \in S$ is assigned the rank $(r_i^k = r_i^j)$ if it corresponds to a feature f_j selected in CFS_i and (n+1) otherwise, i.e. a higher rank than the worst ranked feature in CFS_i. The ranks over all ranking lists are summed up for each feature f_k as follows:

$$Rank(f_k) = \sum_{i=1}^{m} (r_i^k). \tag{2.3}$$

Features with the lowest summed rank are selected as final result Abeel et al. (2010). This proposed approach is called IB-RA.

2.3.2.4 Feature selection by redundancy elimination

The first step of our proposed approach does not detect redundancy, so the obtained feature set still contains redundant features. In this proposed technique, a second redundancy elimination (RedE) filter is introduced where mutual information between each pair of attributes is taken into consideration in order to detect redundant features and overcome this problem. We refer to this approach as IB-RedE.

The relevance score of each feature f_k in S is calculated as the sum of its weights $w_{k,i}$, obtained as explained in the WMA method, over the m instances. The relevance score is given by:

$$Rel(f_k) = \sum_{i=1}^{m} (w_{k,i}). \tag{2.4}$$

Then, the mutual information of two features f_k and f_t is defined in terms of their probabilistic density functions as follows:

$$MI(f_k, f_t) = \int p(f_k, f_t) \log \frac{p(f_k, f_t)}{p(f_k)p(f_t)} df_k df_t. \tag{2.5}$$

The redundancy score of each feature, is then measured by averaging its mutual information with all the features in S, and is given by:

$$Red(f_k) = \frac{\sum_{f_t \in S} MI(f_k, f_t)}{\mid S \mid}. \tag{2.6}$$

After the redundancy score computation, the final score of each feature is calculated based on its relevance and redundancy scores, and is defined as follows:

$$Score(f_k) = Rel(f_k) - Red(f_k). \tag{2.7}$$

Features with the highest scores are selected. In this technique, we use the mutual information based filter to only detect features' redundancy. Nevertheless, it can be also used to find features' relevancy with the target class as it is the case in the minimum Redundancy Maximum Relevance algorithm (mRMR) Peng et al. (2005).

2.3.3 Experimental study

In this section we report the experimental setup and results of our proposed filter methods and comparison results with four existing methods, namely Relief Kira and Rendell (1992), mRMR Peng et al. (2005), t-test and entropy filter methods. The KNN and SVM classifiers are used with all algorithms to evaluate classification performance. Our experimental data consists of seven cancer diagnosis microarrays data sets described in Section (4.1). Classification performance, stability and final subset cardinality are used as metrics to evaluate our approaches.

2.3.3.1 Evaluation

For the proposed methods, the number of final selected features depends on the algorithm setting, i.e. the candidate subset cardinality and the training data. Given that we use 10-fold cross validation to select features and then record their corresponding misclassification error on each test fold, the number of selected features may vary. For each fold, we record classification performance corresponding to all possible feature subset sizes, i.e. for example if 40 features are selected by an approach, classification performance is tested for up

to 40 features. Then, to have a general approximation of the optimal number of features to select, we focus only on the 10-fold shared feature subset sizes and calculate the 10-fold cross validation MCE for each possible cardinality. Stability is also calculated for the cardinality of the optimal subset over the 10 folds.

For candidate feature subsets construction, we evaluate subset cardinalities ranging from 1 to 15 features and record the obtained SFS size and the corresponding MCE.

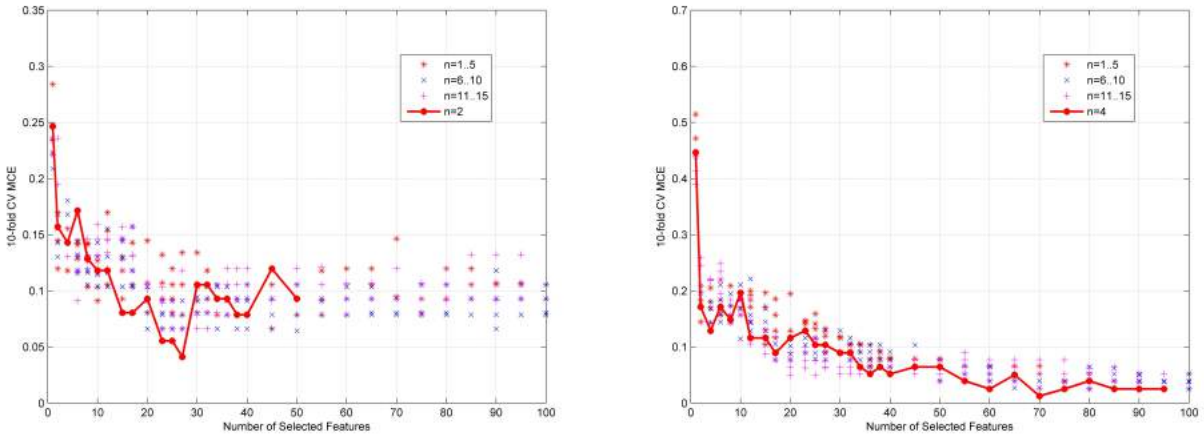
Figures (2.2) - (2.8) show the MCE of IB-OF with kNN and SVM classifiers for the seven microarray data sets. There are three different colors corresponding to three CFS initial intervals presenting the 15 tested cardinalities ([1..5], [6..10] and [11..15]). The curve corresponding to the CFS cardinality that gives the optimal MCE achieved is highlighted and shown by a continuous curve. We can notice that red color corresponding the the first CFS cardinalities interval [1..5] is often present in the figures to show the optimal MCE. We observe also that the red color corresponds to smaller sizes of SFSs. The blue color [6..10] is following leading us to deduce that best performances are often achieved with one of the five or ten first CFS cardinalities. For example, Figure 2.2 shows all obtained MCE of KNN and SVM with IB-OF, for the DLBCL data set, using different initial CFS settings.

For this setting, red color corresponding to the first CFS cardinalities interval [1..5] is used. More specifically, the initial CFSs size, i.e. 2, leads to the selection of 50 features, 27 out of them give the best classification performance with KNN classifier. CFSs of 4 features lead to the selection of 95 features, out of which 70 features give the best classification performance with SVM classifier.

Table 2.3 and 2.4 show CFS cardinalities, used by the four proposed feature selection approaches, that lead to the best classification performances of KNN and SVM classifiers for all data sets. CFS sizes reported often do not exceed 10 features.

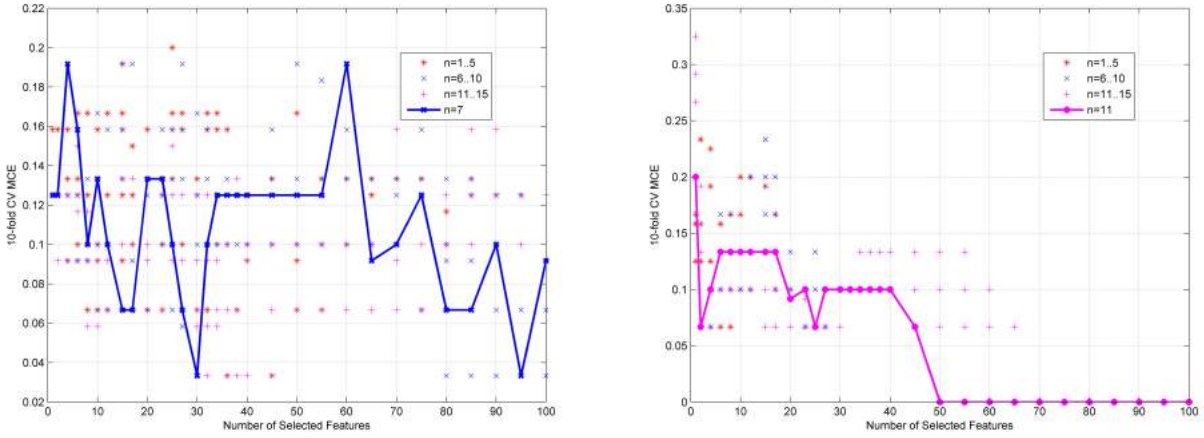
2.3.3.2 Results and comparison with existing algorithms

In this section, we report comparison results of our filter methods with four well known filters, Relief Kira and Rendell (1992), mRMR Peng et al. (2005), t-test and Entropy based feature selection algorithm. The KNN and SVM classifiers are used with all setups to



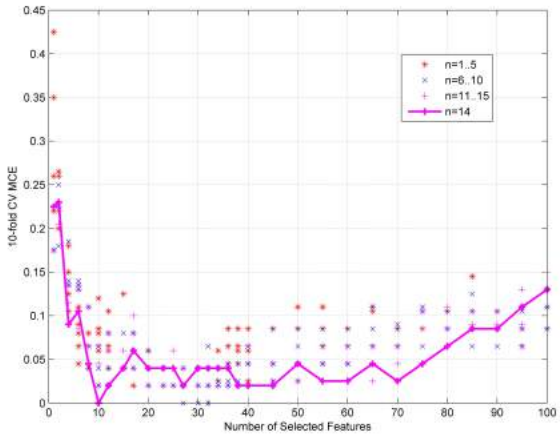
(a) kNN MCE (b) SVM MCE

Figure 2.2: IB-OF MCE for DLBCL data set.

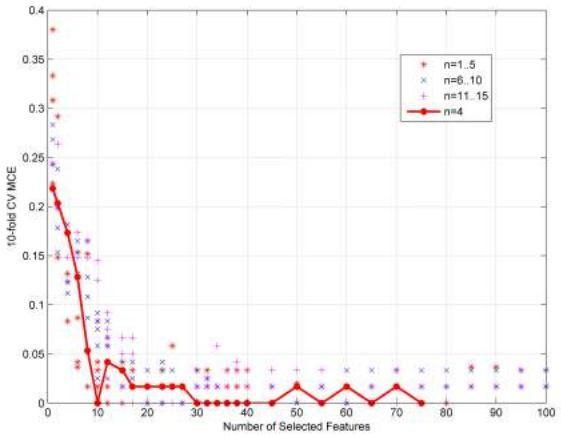


(a) kNN MCE (b) SVM MCE

Figure 2.3: IB-OF MCE for Bladder cancer data set.

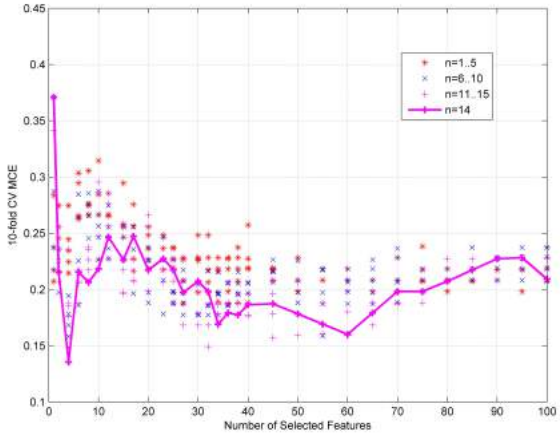


(a) kNN MCE

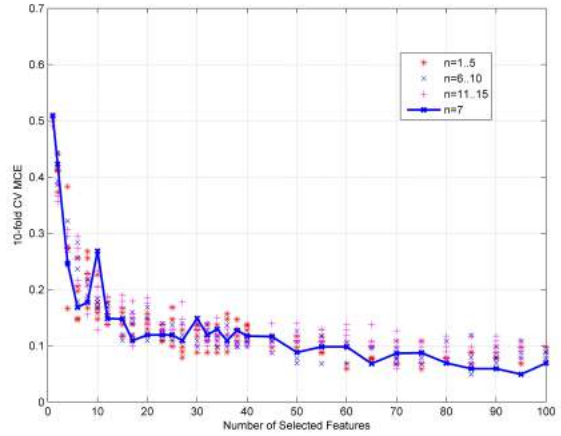


(b) SVM MCE

Figure 2.4: IB-OF MCE for Lymphoma data set.

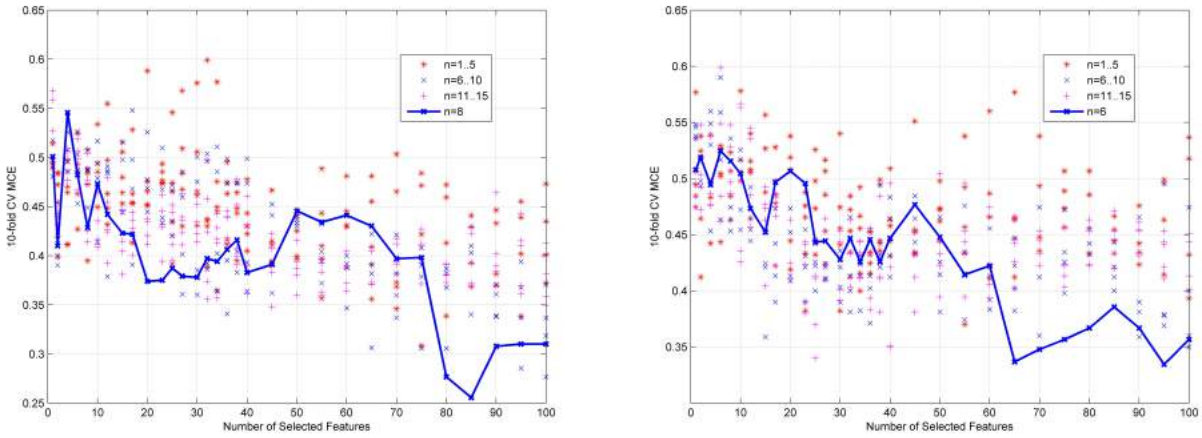


(a) kNN MCE



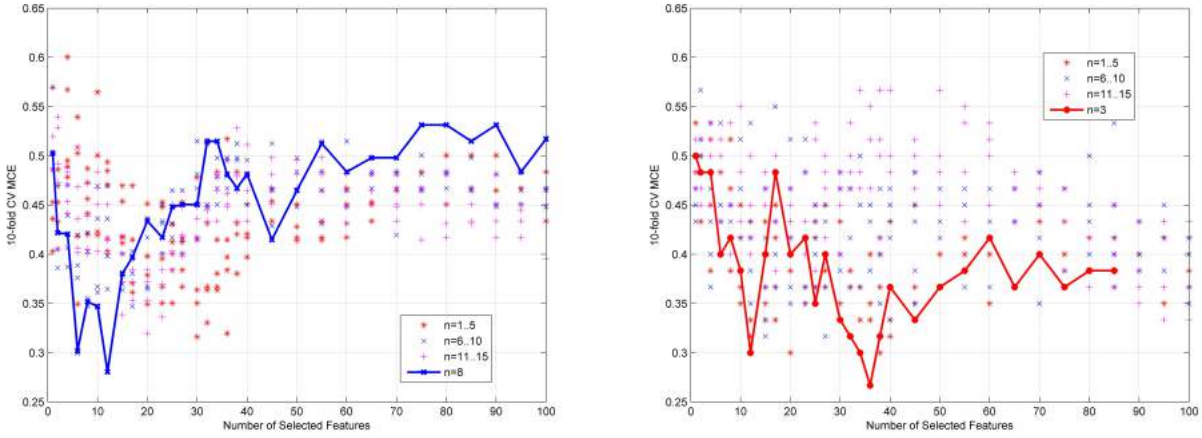
(b) SVM MCE

Figure 2.5: IB-OF MCE for Prostate data set.



(a) kNN MCE (b) SVM MCE

Figure 2.6: IB-OF MCE for Breast data set.



(a) kNN MCE (b) SVM MCE

Figure 2.7: IB-OF MCE for CNS data set.

Table 2.3: CFS size that leads to the best performance of KNN classifier.

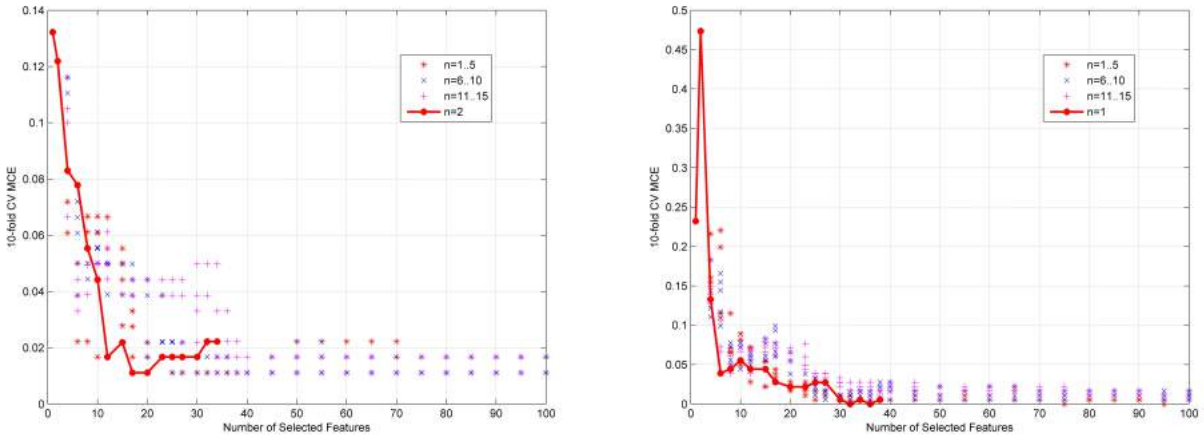
	IB-OF	IB-WMA	IB-RA	IB-RedE
DLBCL	2	7	3	6
Bladder	7	7	9	11
Lymphoma	14	8	8	3
Prostate	14	14	15	10
Breast	8	1	2	15
CNS	8	2	2	5
Lung	2	2	4	3
AVG	7.85	5.85	6.14	7.57

Table 2.4: CFS size that leads to the best performance of SVM classifier.

	IB-OF	IB-WMA	IB-RA	IB-RedE
DLBCL	4	8	11	13
Bladder	11	8	11	3
Lymphoma	4	1	1	2
Prostate	9	4	9	10
Breast	6	5	12	6
CNS	3	1	4	9
Lung	1	15	3	1
AVG	5.42	6	7.28	6.28

evaluate classification performance. The considered algorithms are applied to several microarray data sets described in Section 4.1. As for our filter methods and to avoid having a local minimum of the cross-validation MCE, we tested the performance of algorithms as a function of the number of features for up to 100 features and recorded the minimum MCE rate and the corresponding SFS cardinality for each algorithm. Stability of the SFS is also used as metric to evaluate and compare our approaches. Table 2.7 shows results obtained with KNN classifier for the seven microarray data sets.

In terms of classification performance, we observe that the proposed IB algorithms are competitive and often outperform other filters, specially IB-OF and IB-RedE that give the best performances. Classification results of these two algorithms are similar in many cases except for Prostate and Breast cancer data sets. IB-RedE that uses a redundancy filter,



(a) kNN MCE

(b) SVM MCE

Figure 2.8: IB-OF MCE for Lung cancer data set.

achieves a MCE of 7% on Prostate cancer data set, outperforming IB-OF and other proposed algorithms with a significant difference. This difference is less important compared to mRMR algorithm achieving a MCE of 7.84% for the same data set. For Breast cancer data set, IB-OF based on feature occurrence frequency, is the method giving the best classification performance with a MCE difference of more than 5%. It is however less stable than the other proposed algorithms for this data set and it gives the worst stability result for Prostate cancer data set. IB-WMA and IB-RA filters give also good classification results which are also competitive with a highest MCE difference of about 4%, in favor of IB-WMA, for CNS data set. In terms of stability results, IB-WMA and IB-RA are often more stable than the two other proposed IB filters. IB-RA is however less stable for Prostate data set (64.25%), and IB-WMA yields a poor feature selection stability (39.99%) for CNS data set. IB-RedE and IB-OF are generally less stable. Nevertheless, all the proposed IB filters are still more stable than Relief and mRMR algorithms.

Relief algorithm gives modest results for both classification performance and stability. Among existing feature selection methods, the mRMR algorithm is characterised by the selection of relevant features while removing redundancy, resulting on good classification performance in many cases. However, stability results of mRMR are modest. For simple filters like t-test and Entropy based filter, classification performance vary depending on the data set considered. They may give good (Bladder, Lung), modest (Prostate, CNS) or poor

Table 2.5: Compared KNN minimum MCE rates, SFS cardinalities and stability on cancer diagnosis data sets.

		IB-OF	IB-WMA	IB-RA	IB-RedE	Relief	mRMR	t-test	Entropy
DLBCL	MCE	**0.0411**	**0.0411**	0.0500	0.0518	0.0649	0.0779	0.1429	0.1429
	# SFS	27	50	12	45	75	45	55	95
	Stab	0.7277	0.8185	0.7776	0.6680	0.6335	0.5657	0.8791	0.8122
Bladder	MCE	**0.0323**	**0.0333**	**0.0333**	**0.0333**	0.0645	0.0645	**0.0323**	**0.0323**
	#SFS	30	12	10	15	60	20	12	10
	Stab	0.6906	0.6627	0.6883	0.7688	0.4650	0.4039	0.6877	0.7057
Lymph	MCE	**0**	0.0250	**0**	**0**	0.0222	**0**	0.0222	0.0667
	# SFS	10	10	32	20	30	12	15	32
	Stab	0.6142	0.7107	0.7026	0.6703	0.4291	0.5951	0.7234	0.7578
Prostate	MCE	0.1355	0.1482	0.1573	**0.0700**	0.2157	0.0784	0.1078	0.0980
	# SFS	4	60	30	60	38	25	4	30
	Stab	0.4443	0.7493	0.6425	0.7387	0.6275	0.7096	0.7944	0.6852
Breast	MCE	**0.2555**	0.3111	0.3062	0.3073	0.4433	0.2784	0.5258	0.5258
	# SFS	85	60	90	80	6	30	2	2
	Stab	0.4231	0.5097	0.5082	0.4611	0.3628	0.3355	1.0000	1.0000
CNS	MCE	**0.2805**	0.2986	0.3367	0.2833	0.3667	0.3000	0.3667	0.3333
	# SFS	12	1	6	8	90	50	25	4
	Stab	0.5138	0.3999	0.5275	0.5508	0.4974	0.3792	0.3970	0.5664
Lung	MCE	0.0110	0.0056	0.0108	**0.0053**	0.0110	0.0055	0.0166	0.0055
	# SFS	17	15	27	20	20	4	15	17
	Stab	0.8567	0.8721	0.8990	0.6575	0.8409	0.6610	0.8368	0.8900
AVG	MCE	0.1081	0.1232	0.1277	0.1072	0.1697	0.1149	0.1736	0.1722
	# SFS	26.42	29.71	29.57	35.42	45.57	26.57	18.28	27.14
	Stab	0.61	0.6747	0.6779	0.645	0.5537	0.5214	0.7597	0.77

(DLBCL, Breast) classification results when compared with other algorithms. Especially for Breast data set, the MCE exceeds 50% for the two filters. However, for both filters, stability is perfect (100%) for the selected features that yielded this poor performance. Most often, high stability is required for good feature selection. However in this case, it is coupled with poor classification performance and as it was argued, stability of feature selection is not enough but it should be considered together with classification accuracy, because domain experts are not interested in a strategy that yields very stable feature sets, but leads to a bad predictive model. Accordingly, t-test and Entropy methods are not reliable for data sets for which they give good feature selection stability but high MCE.

Also, we compared our approaches with the four algorithms based on SVM classifier results. This gives us the possibility to test the generalization capabilities of all considered algorithms and to know if we can obtain the same conclusions concerning their results,

especially classification accuracy, for different classifiers. Table 2.8 shows SVM results for the eight feature selection approaches.

Table 2.6: Compared SVM minimum MCE rates, SFS cardinalities and stability on cancer diagnosis data sets.

		IB-OF	IB-WMA	IB-RA	IB-RedE	Relief	mRMR	t-test	Entropy
DLBCL	MCE	**0.0125**	0.0268	0.0375	0.0375	0.0390	0.0519	0.1299	0.1818
	# SFS	70	55	45	70	70	75	85	85
	Stab	0.7779	0.8521	0.8017	0.6958	0.6627	0.6089	0.8476	0.8164
Bladder	MCE	**0**	0.0333	0.0333	0.0333	0.0968	0.0968	0.0968	0.0968
	# SFS	50	50	23	34	38	95	20	10
	Stab	0.7347	0.7091	0.8074	0.6747	0.4445	0.5982	0.7841	0.6923
Lymph	MCE	**0**	**0**	**0**	**0**	0.0444	0.0222	**0**	**0**
	# SFS	10	8	6	8	21	20	27	75
	Stab	0.6958	0.7108	0.6107	0.6796	0.4490	0.6460	0.7986	0.7509
Prostate	MCE	0.0491	0.0582	0.0600	**0.0400**	0.0784	0.0784	0.0980	0.1176
	# SFS	85	60	60	85	65	65	45	65
	Stab	0.7646	0.7774	0.7483	0.7415	0.5838	0.7718	0.8365	0.6722
Breast	MCE	0.3344	0.3045	0.3308	0.3175	0.4021	**0.2680**	0.5052	0.5155
	# SFS	95	90	95	100	80	10	2	2
	Stab	0.5246	0.5711	0.5083	0.3612	0.2609	0.3086	1.0000	1.0000
CNS	MCE	0.2667	0.2819	0.2962	**0.1952**	0.3500	0.3667	0.3500	0.2833
	# SFS	36	15	32	100	80	65	23	75
	Stab	0.6135	0.5189	0.6051	0.5327	0.5112	0.4442	0.4223	0.7092
Lung	MCE	**0**	**0**	**0**	0.0056	**0**	0.0055	0.0166	0.0055
	# SFS	32	50	85	30	80	50	80	8
	Stab	0.9006	0.9247	0.8934	0.4518	0.7702	0.8193	0.8831	0.8944
AVG	MCE	0.0946	0.1006	0.1082	0.0898	0.1443	0.127	0.1708	0.1715
	# SFS	54	46.85	49.42	61	62	54.28	40.28	45.71
	Stab	0.7159	0.7238	0.7107	0.591	0.526	0.5995	0.796	0.79

According to average results, when used with SVM classifier, our algorithms give slightly better results than KNN in most cases but have the same behaviour. SVM classification result with IB-RedE on CNS data set is significantly better than KNN whith an accuracy improvement of about 9%. Also, for Prostate cancer data set, classification performances of SVM using each of our proposed approaches and Relief algorithm, are far much better than KNN results. However, the opposite is noticed for Breast cancer data set, especially with IB-OF giving a MCE of about 33% with SVM, i.e. worse than KNN of about 8%. Nevertheless, the experimented algorithms have the same behaviour with SVM classifier for most data sets where IB-OF and IB-RedE still are the most efficient in terms of classification accuracy. This is expected from filters which select features independently of the classification algorithm. Thus, they are likely to have a good generalization ability.

This result is confirmed by our experiments where we deduce that variations in classification results are due to classifier's properties.

In addition to classification performance and stability, it is interesting to compare selected feature subset cardinality of the experimented approaches. In some cases, a small number of selected features gives better results than large subsets of relevant features selected by other approaches. However, it is not always the case. For example, KNN best performance on Breast cancer data set $(25, 55\%)$ is obtained with 85 features selected by IB-OF. mRMR gives a MCE of about 27% with only 30 features. Since it is always better to have a minimum number of selected features, mRMR may be preferred for Breast cancer data set if this evaluation criterion is considered. For the same data set, and with SVM classifier, the same thing is observed concerning selected feature subset cardinality. However, this time the classification performance is in favor of mRMR algorithm with only 10 features. For CNS data set, SVM with 100 features selected with IB-RedE yields the best classification performance which is far much better than other algorithms. In this case, the SFS high cardinality may not be a problem if it results on a very good classification result. Thus, selected feature subset cardinality importance depends on criteria preferences of the domain expert. Results show also that stability of feature selection is not affected by the SFS size.

2.3.4 Discussion

To summarize, the proposed IB filters are the MCE-stability optimization challenge winners. IB-OF and mainly IB-RedE, based on redundancy elimination, are specially favored if classification accuracy is preferred to stability. IB-WMA and IB-RA are more efficient in terms of stability and have a similar behaviour. The reason is that IB-WMA aggregates feature's weights and IB-RA aggregates feature's ranks which are precisely obtained by ranking their weights. Performances of existing feature selection methods vary depending on the data set considered and they generally show good classification or good stability performance, but not both on the same time, making them less reliable than our approaches.

In the next section, we investigate research on using small sample size to create hybrid feature selection methods which take as input CFSs, obtained as described in the filter methods described above, and involve them in a wrapper process for the optimal feature

subset research. The proposed hybrid approaches will take advantage of filter but also wrapper's strengths.

2.4 Hybrid instance based feature selection algorithms

In practice, for high dimensional data, it is very common to use filter methods that measure the strength of relationship between each gene and the class label. However, Tolosi and Lengauer (2011) demonstrated that filters ignore the correlations between genes, which are prevalent in gene expression data due to gene co-regulation. The consequence is that many redundant differentiated genes are included, meanwhile, useful but weakly differentiated genes may be omitted. On the other hand, Kohavi and John (1997) showed that standard wrapper algorithms cannot be applied because of their high computational complexity due to the need to train a large number of classifiers. With tens of thousands of features, which is the case in the studied gene expression microarray data sets, a hybrid approach can be adopted. It should follow a filter model in the search step selecting small number of CFSs. Then, a wrapper method is applied to the reduced subsets to achieve the best possible performance with a particular learning algorithm. Accordingly, the hybrid model is expected to be more efficient than filter and less expensive than wrapper.

In this second part of the chapter, we propose new hybrid approaches for feature selection on cancer diagnosis data sets. In these approaches, each instance is an expert which proposes a CFS based on an instance feature weighting technique. The CFSs are then integrated in a search procedure of the optimal feature subset, where sequential backward search (SBS) and cooperative subset search (CSS), are proposed as two alternatives used with kNN as evaluation systems of wrappers. The main goal of our proposed methods is to speed up the feature subset selection process by reducing the number of wrapper evaluations while maintaining good performance in terms of accuracy and size of the obtained subset.

Our hybrid feature selection methods begin by a filter step where each instance is an expert which proposes a CFS based on an instance feature weighting technique. This technique is the same used in the IB-filter proposed in the presvious section of the chapter. So we only detail the second phase of the proposed hybrid approaches. In the second step, the CFSs are integrated in a search process of the optimal feature subset, where the subset

search technique and kNN classifier consist an evaluation system of wrappers. For this step, we propose two alternatives, SBS and CSS algorithms. For both algorithms, the best feature subset search technique is done with ten-fold CV. The two hybrid approaches are illustrated in Figure (2.9) and their two step process is detailed in the following subsections.

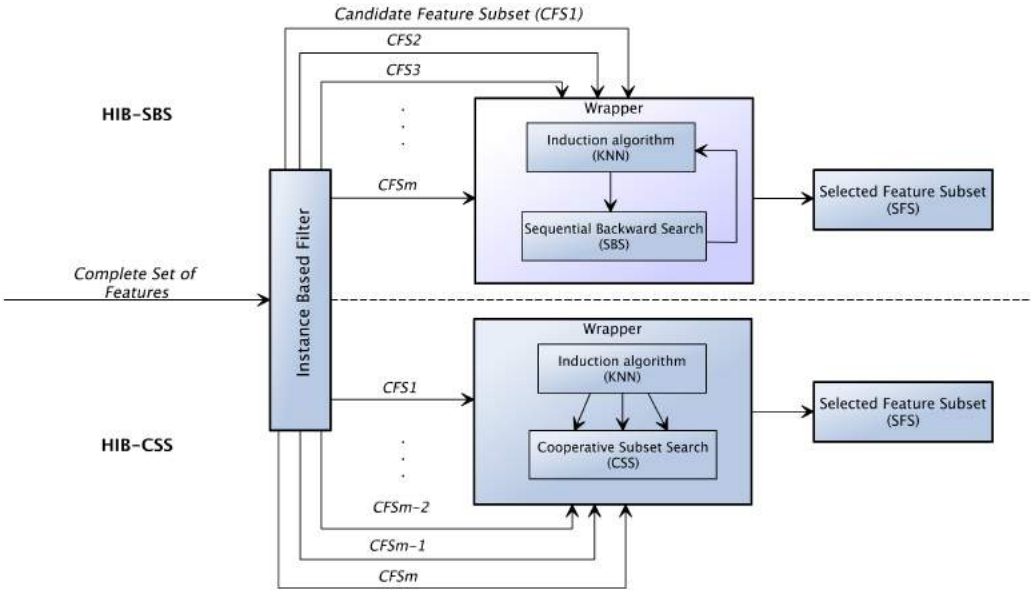


Figure 2.9: Hybrid Instance Based Feature Subset Search

2.4.1 First wrapper alternative: SBS

This step begins by considering all the features composing the m CFS in a single feature subset called FS_{Union}. A kNN classifier is trained on the projection of FS_{Union} on the training data and the classification error β_{init} is calculated on the test data set using 10-fold CV. Then, a kNN classifier is applied on the training data using $FS_{Union} \setminus \{CFS_i\}$ and its classification error rate β_i is calculated. Thus, the kNN classifier is applied m times and m classification error rates $ERR = \{\beta_1, \beta_2, ...\beta_m\}$ are obtained. The algorithm then finds in ERR the error rates which are smaller than β_{init}. The resulting error rates subset is $ERR\prime \subset ERR$. If $ERR\prime$ is empty, FS_{Union} is selected as the final feature subset as it gives the minimum error rate. If it is not the case, candidate subsets which their exclusion has resulted in error rates of $ERR\prime$ are rejected and FS_{Union} is updated to contain features of remaining CFS. That means that the K worst CFS are eliminated from the wrapper search

procedure. Thus, the number of CFS decreases to $(m - K)$. The whole error β_{init} is also updated to be equal to β_i, the minimum classification error in $ERR\prime$. Hence, the optimal feature subset to be selected should reduce β_i the minimum error rate achieved in the last iteration. This SBS process is iterated until there is no decrease in the classification error rate, i.e. until $ERR\prime$ is empty. The resulting feature subset FS_{Union} is returned as the optimal feature subset. Note that 10-fold CV is used for the feature subset search. We call this version of our proposed approach Hybrid Instance Based SBS (HIB-SBS) and its algorithm is described in Algorithm 1.

Algorithm 1 HIB-SBS

Input:

$[\mathbf{X}, CFS_{All}]$

Set $FS_{Union} = \forall f_j \in CFS_{All}$

β_{init} = Apply kNN (X, FS_{Union})

repeat

β_i = Apply kNN $(X, (FS_{Union} \setminus \{CFS_i\}))$

Obtain $ERR = \{\beta_1, \beta_2, ...\beta_m\}$

Obtain $ERR\prime = \forall \beta_i < \beta_{init}$

if $ERR\prime = \emptyset$ return selected features FS_{Union}

else

Find "Bad CFSs" : the K CFSs resulting in $\beta_i \in ERR\prime$

Update $CFS_{All} = (CFS_{All} \setminus$ Bad CFSs)

Update $FS_{Union} = \forall f_j \in CFS_{All}$

Update $m = m - K$

$\beta_{min} = min(ERR\prime)$

Update $\beta_{init} = \beta_{min}$

until $ERR\prime = \emptyset$

Output: FS_{Union}

2.4.2 Second wrapper alternative: CSS

This wrapper approach is based on CSS, i.e feature selection decisions of training instances are combined based on their effect on classification performance without using an iterative process, but in a parallel manner instead. Feature weights obtained in the filter step are given as inputs to the wrapper approach. The value of $w_{j,k}$ depends on whether the feature f_j appears or not in the candidate subset CFS_k. Thus, feature weights will take the following values :

$$w_{j,k} = \begin{cases} w_{j,k} & \text{if } w_{j,k} \in CFS_k \\ 0 & \text{Otherwise.} \end{cases}$$

In the CSS process, a kNN classifier is trained on the projection of each CFS_i on the training data and the classification error β_i is calculated on the test set data using 10-fold CV. Thus, the kNN classifier is applied m times and m classification error rates are obtained. Two thresholds ε_{min} and ε_{max} are used to fix the good CFS and the bad CFS, such as CFS_i is good if its corresponding classification error β_i is less than ε_{min}, and it is bad if its corresponding classification error β_i is higher than ε_{max}. Based on this categorization, we obtain FS_{Good}, the subset containing the features appearing in the good CFS, and FS_{Bad} which is the subset containing the features appearing in the bad CFS. The K CFS that correspond to the K error rates which are smaller then ε_{max} are called "Other CFS". Their component features are gathered in a single feature subset called FS_{Union} and their feature weights are updated based on their corresponding error rates. This weight adjustment aims to penalize more features of CFSs that result in higher classification error rates in "Other CFS" group. Then, the total weight of each feature in FS_{Union} is calculated as the aggregated sum of its weights over candidate subsets in "Other CFS". Based on the calculated weights, features in FS_{Union} are ranked and the S best features are selected. This SFS is updated in two steps. In the first step, features of FS_{Bad} are extracted from FS_{Union}. In the second step, features of FS_{Good} are added to FS_{Union}. The resulting feature subset FS_{Union} is returned as the optimal feature subset. Note that the pre-selection step of S feature size subset can be omitted and replaced by the test of several feature subset cardinalities once the final feature subset is obtained. The feature subset size that gives the best classification performance is chosen as it will be seen in the experimental study of the

algorithm in Section 5. HIB-CSS algorithm is reported in Algorithm 2.

Algorithm 2 HIB-CSS

Input:

$[\mathbf{X}, CFS_{All}, W, \varepsilon_{min}, \varepsilon_{max}]$

β_i = Apply kNN (X, CFS_i)

Obtain $ERR = \{\beta_1, \beta_2, ...\beta_m\}$

Find "Good CFSs" = CFSs corresponding to $\beta_i < \varepsilon_{min}$

Obtain $FS_{Good} = \forall f_j \in$ "Good CFSs"

Find "Bad CFSs" = CFSs corresponding to $\beta_i > \varepsilon_{max}$

Obtain $FS_{Bad} = \forall f_j \in$ "Bad CFSs"

Find "Other CFSs" corresponding to $\beta_i < \varepsilon_{max}$

K is the number of CFS in "Other CFSs"

Obtain $FS_{Union} = \forall f_j \in$ "Other CFSs"

Update weights of features $f_j \in FS_{Union}$:

$w_{f_j} = \sum_{k=1}^{K} (\frac{w_{j,k}}{\beta_k})$

Rank features in FS_{Union} based on w_{f_j}

Keep S best ranked features in FS_{Union}

Update $FS_{Union} = (FS_{Union} \setminus FS_{Bad})$

Update $FS_{Union} = (FS_{Union} \cup FS_{Good})$

Output: FS_{Union}

2.4.3 Experimental study

In this section we report the experimental setup and results of our proposed hybrid feature selection methods and comparison results with four existing methods. The kNN classifier is used with all algorithms to evaluate classification performance. The considered algorithms are applied to several microarray data sets described in Section 2.2.1. Classification performance, final subset cardinality and execution time are used as metrics to evaluate our approaches.

2.4.3.1 Evaluation

We use 10-fold stratified CV to predict the classification performance and stability of kNN algorithm in the sequential search procedures on seven data sets. We evaluated also the final SFS cardinality obtained and the execution time (in seconds) to compare our proposed hybrid methods to other existing feature selection methods.

2.4.3.2 Performance of proposed algorithms

From the first chapter contribustion experiments, we deduced that experimenting ten initial CFS cardinalities ranging from 1 to 10 features is sufficient to have an optimal performance. So we experimented our proposed hybrid approaches with this setting. For HIB-SBS, the SFS cardinality is obtained by the algorithm. Thus, for the obtained cardinality, we tested all subset sizes possibilities with the classification algorithm. Performance of HIB-CSS with SFS cardinality for up to 100 features was tested. We recorded the MCE and the corresponding SFS cardinality for each setting of the two proposed hybrid methods. Table (2.7) shows the results of the application of our proposed hybrid methods on the seven data sets. Minimum MCE (Min MCE) obtained and the corresponding CFS and SFS cardinalities are reported.

Table 2.7: HIB-SBS and HIB-CSS results on cancer diagnosis data sets.

	HIB-SBS			HIB-CSS		
	# CFS	# SFS	Min MCE	# CFS	# SFS	Min MCE
DLBCL	9	31	0.0518	5	28	0.0500
Bladder	3	46	0.0917	9	17	0.0333
Lymphoma	10	5	0.0200	5	7	0.0000
Prostate	6	25	0.1182	2	25	0.0991
Breast	4	17	0.3067	9	100	0.2889
CNS	6	11	0.2486	1	1	0.2319
Lung	8	87	0.0108	5	129	0.0000

For all data sets, HIB-CSS classification results are better that those obtained with HIB-SBS. Indeed, HIB-CSS achieved a best MCE of 0% (100% accuracy) on two data sets, Lymphoma and Lung cancer. For this algorithm, SFS cardinalities range from 1 to 28 selected features for five data sets. However, for the two others, it is in the neighborhood

of 100. Nevertheless, if a small SFS is preferred, one can sacrifice a small classification performance. For example results obtained give a MCE of $0,53\%$ with only 11 features and 1 CFS cardinality for Lung cancer data set. So in this case, one can choose the SFS cardinality to work with based on its performance priority : optimal MCE or optimal SFS cardinality. It is important to notice that only 1 feature selected by HIB-CSS gives the best MCE for CNS data set, using a minimal CFS cardinality of only 1 feature also, as initial setting.

For HIB-SBS, 87 is the maximum SFS cardinality used with Lung cancer data set to obtain $1,08\%$ MCE. The SFS ranges between 5 and 46 features for the six other data sets. Often, initial CFS setting ranges between 4 and 6 features.

2.4.3.3 Comparison with other algorithms

In this section, we report comparison results of our proposed methods and four feature selection methods, Relief and t-test algorithms which are filters, Randomized which is a wrapper and another hybrid algorithm. The three first algorithms are described in Chapter 1. The hybrid algorithm used for our comparisons in addition to these algorithms uses a forward sequential feature selection with kNN algorithm in a wrapper fashion. It finds important features from a reduced set of features obtained using filter results of a t-test as a pre-processing step. We refer to this algorithm in our experiments as "Hyb-Seq". The kNN classifier is used with all setups to evaluate classification performance. The considered algorithms are applied to several microarray data sets described in Section 5.1. As for HIB-CSS and to not have a local minimum of the 10 CV MCE, we tested the performance of algorithms as a function of the number of features for up to 100 features and recorded the optimal MCE rate and the corresponding SFS cardinality for each algorithm. The execution time (in seconds) is also used as metric to evaluate and compare our approaches.

Comparative Results: We report in Table (2.8) the best classification results, the corresponding feature subset cardinality and stability. To have a clearer vision on the results, we underlined the two best classification performances (MCE) and stabilities for each data set. Let's remember that algorithms that give best couple MCE-stability results are considered good feature selection algorithms.

For six out of seven data sets, HIB-CSS is among the two best algorithms in terms of

MCE results. It is in the first place in four cases out of six. And finally, HIB-CSS gives the best MCE-stability performance for three data sets (DLBCL, CNS and Lung cancer). For the same data sets, HIB-SBS gives the second best MCE following HIB-CSS. However, it is not performing well in terms of stability and this is an important shortcoming for HIB-SBS.

For Bladder and Lymphoma data sets, Hyb-Seq algorithm gives a best MCE-stability couple performance. It gives minimum MCE for Prostate cancer data set coupled with poor stability (34%) and a very bad MCE (52%) coupled with a perfect stability (100%) for Breast cancer data set. Thus, the trade-off MCE-Stability is not satisfied in these cases.

Best stability is often achieved by t-test filter. However, in most cases the trade-off we are interested in is not satisfied here again as t-test gives bad MCE results for some data sets (14% and 52% MCE respectively for DLBCL and Breast cancer data sets). Finally, for only one data set (Breast cancer), Randomized algorithm achieves the optimal MCE followed by HIB-CSS algorithm, and Relief filter is not showing special good results.

To summarize, HIB-CSS is the MCE-stability optimization challenge winner. It is followed by Hyb-Seq algorithm. HIB-SBS and t-test performances do not satisfy the trade-off. HIB-SBS gives good classification performance but stability is its shortcoming, and vice versa for t-test filter which classification performance deteriorates completely for some data sets making it unreliable.

The execution time (in seconds) of our proposed methods and the four other state of the art methods is reported in Table (2.9). It is noticeable that HIB-SBS and HIB-CSS are close to each other. Their execution times are higher than Hyb-Seq algorithm for many case, but often extremely smaller than the randomized feature selection which is a wrapper approach. Filters still are the fastest algorithms with smallest execution times achieved by the t-test filter. This is expected as filters select features independently of the classifier and thus avoid the CV step used in the wrapper and hybrid algorithms. However, it is important to notice that the running time for both HIB-SBS and HIB-CSS algorithms includes ten initial settings testing time corresponding to different CFS cardinality settings in the wrapper step.

2.4.4 Discussion

To ovoid overfitting of the data, feature selection is required when the number of features is large with respect to the sample size. Filter feature selection methods are well suited to

Table 2.8: MCE rates, SFS cardinalities and stability on cancer diagnosis data sets.

		HIB-SBS	HIB-CSS	Hyb-Seq	Randomized	t-test	Relief
DLBCL	Min MCE	0.0518	0.0500	0.1039	0.0519	0.1429	0.0649
	#SFS	31	28	10	40	55	65
	Stability	0.3095	0.6893	0.2078	0.0709	0.8791	0.6335
Bladder	Min MCE	0.0917	0.0333	0.0323	0.0645	0.0323	0.0645
	#SFS	46	17	36	38	12	60
	Stability	0.5880	0.5925	0.7389	0.0667	0.6877	0.4650
Lymphoma	Min MCE	0.0200	0.0000	0.0000	0.0667	0.0222	0.0222
	#SFS	5	7	27	30	15	30
	Stability	0.2213	0.2336	0.7158	0.0686	0.7234	0.4291
Prostate	Min MCE	0.1182	0.0991	0.0686	0.0882	0.1078	0.2157
	#SFS	25	25	45	87	4	38
	Stability	0.2251	0.6553	0.3423	0.0216	0.7944	0.6275
Breast	Min MCE	0.3067	0.2889	0.5258	0.2680	0.5258	0.4433
	#SFS	17	100	1	75	2	6
	Stability	0.2662	0.4718	1.0000	0.0168	1.0000	0.3628
CNS	Min MCE	0.2486	0.2319	0.3167	0.3333	0.3667	0.3667
	#SFS	11	1	4	17	25	90
	Stability	0.2089	0.6222	0.1217	0.0448	0.3970	0.4974
Lung	Min MCE	0.0108	0.0000	0.0276	0.0552	0.0166	0.0110
	#SFS	87	129	8	80	15	20
	Stability	0.5126	0.8804	0.1217	0.0112	0.8368	0.8409

such applications as they are fast. However, they ignore the correlations between features and their interaction with the learning algorithm and thus may have modest classification performance. Wrappers on the other hand use the bias of the induction algorithm to select features and generally perform better. However, the computational burden of wrapper methods is prohibitive on large data sets. In this chapter contribution, we proposed two new hybrid approaches, HIB-SBS which is based on feature sequential search and HIB-CSS which is based on cooperative search. The two proposed approaches use instance learning in their filter step. Their main goal is to speed the feature subset selection process by reducing the number of wrapper evaluations while maintaining good performance in terms of accuracy, stability and size of the obtained subset. The main challenge in these approaches is that they convert the problem of the small sample size to a tool that allows choosing only a few subsets of variables to be analyzed since the number of CFSs is the number

Table 2.9: Execution times on cancer diagnosis data sets.

	Running Time (in Sec)					
	HIB-SBS	HIB-CSS	Hyb-Seq	Randomized	t-test	Relief
DLBCL	683.5028	618.9702	606.7253	4.5951e+003	1.1814	7.9377
Bladder	128.5865	119.6286	413.1714	1.4347e+003	0.6522	1.0190
Lymphoma	236.9500	217.9579	537.7784	2.7602e+003	0.7435	1.7639
Prostate	1.5741e+003	1.3972e+003	751.5693	1.2921e+003	1.2049	25.3161
Breast	2.7221e+003	2.6019e+003	729.8408	4.8045e+003	1.6092	64.6571
CNS	538.6874	478.1515	513.5271	7.8686e+003	0.8487	5.3489
Lung	3.1155e+003	2.7986e+003	1.4490e+003	11.0735	1.8769	75.2938

of instances. Therefore, the number of wrapper evaluations decreases significantly. Our methods are experimentally tested and compared with existing feature selection algorithms based on seven high-dimensional low sample size datasets. Results show that HIB-CSS is the MCE-stability optimization challenge winner outperforming compared hybrid, wrapper and filter methods. HIB-SBS is performing well in terms of classification accuracy but does not satisfies the trade-off as stability is its shortcoming. The execution time of the proposed hybrid approaches is similar to existing hybrid method and is extremely smaller than the randomized feature selection which is a wrapper approach.

2.5 Conclusion

In many classification domains, the dimensionality and small sample size of data causes overfitting and other serious problems to machine learning algorithms. Dimensionality reduction is a solution, however none of the existing feature selection algorithms has been conceived to handle the small sample size nature of the data which is also one of the main causes of feature selection instability. For this reason, we investigated research on feature selection approaches which take into account the explained specificity of data. We proposed a filter and two hybrid algorithms based on instance learning. In the proposed methods, each instance proposes a candidate subset of the most relevant features for this instance. Small sample size makes this process feasible with acceptable running time. The proposed filter selects features by simply counting their frequency of appearance in the candidate subsets. By another hand, the proposed hybrid methods employ the predictive power of wrappers to select a final subset based on two search strategies, sequential and

cooperative. The proposed filter presents the best performance. And generally, proposed filter and hybrid methods that use a combination scheme based on consensus feature selection yield the optimal classification-stability performances. This conclusion makes us naturally think of ensemble methods which main concept is the combination of several algorithms' decisions in a consensus manner. The next chapter is dedicated to the conception of ensemble methods for stable feature selection.

Chapter 3

Ensemble Feature Selection

Contents

3.1 Introduction

The principal goal of machine learning is to achieve the best possible classification performance. This purpose is typically accomplished by using a relevant set of features that improves the model generalization. Many feature selection methods are available and we are faced with the problem of selecting the appropriate feature selection method for a given classification problem. Using feature selection algorithms individually may not automatically lead to better performance, because a single feature selection algorithm focuses on one particular region of the feature space. However, different feature selection algorithms will choose different feature subsets. We may not say that a resulting subset is better than the others but rather that all the obtained subsets are the best subsets among the whole feature space.

To deal with this issue, we naturally think of ensemble learning (Dietterich (2000)) as a way to combine independent feature subsets obtained by a function or data perturbation in order to get a robust feature subset.

The fusion of different features selectors is a step to generate a new feature set from the individual selected sets of features. There are two possible alternatives to combine the results of multiple feature selection algorithms for classification problems which have been proposed in literature. These two alternatives are based on two aggregation levels, classifier aggregation level and selector aggregation level. In the first level, different feature subsets are generated and used for constructing an ensemble of accurate and diverse base classifiers. Classifiers' outputs are then combined to obtain the final classification results. The second aggregation level finds a consensus between the results obtained by several feature selection methods in order to obtain a unique feature subset before the classification process. These two levels of ensemble feature subsets aggregation are detailed below. In this chapter, we propose a feature selection framework that fuses the results obtained by different selection methods. We investigate the effect of ensemble feature selection on the model accuracy by looking deeply into the ensemble feature selection, and performing a comparative study between the performance related to classifier aggregation level and selector aggregation level.

Since feature selection stability is as important as classification accuracy, we are interested on having a single and combined feature subset. Thus, we focus on promoting ensemble feature selection at the selector aggregation level. Hence, we propose an ensemble feature selection approach based on a robust feature aggregation technique to combine the feature selection ensemble. In this approach, simplicity and fastness of filters is used to select the best feature subsets among the whole feature space. Then, the ability of a classification algorithm to provide an associated classification performance is exploited to guide the choice of a final robust subset among initial feature subsets.

3.2 A comparative study on ensemble feature selection aggregation levels

3.2.1 Ensemble learning

Ensemble learning, discussed by Dietterich (2000), consists of constructing a set of classifiers, such as decision trees or neural networks, for the same original problem. To classify a new instance, decisions of single classifiers are combined by voting or averaging leading to a more accurate classification decision. This process imitates human's second nature to consult several persons before making a final decision in different life domains such as medicine, finance, social problems, etc. In machine learning, this process has been extended to improve the performance of the decision making in many domains like bioinformatics, remote sensing, manufacturing, geography, information security, information retrieval and image retrieval. Ensemble methods have enjoyed success and popularity since two decades. The progress started when Schapire (1990) introduced the idea of Boosting the low accuracy of a weak learning algorithm that performs only slightly better than random guessing in the probably approximately correct learning model. He showed that a strong classifier can be obtained by combining an ensemble of weak classifiers. At the same time, Hansen and Salamon (1990) introduced the method of running an ensemble of neural networks trained on the same database and combined using a consensus scheme. The example of a popular combination rule is the ordinary majority voting, where the winning class is determined by the simple majority.

3.2.2 Ensemble Feature Selection

Ensemble feature selection techniques use an idea similar to ensemble learning for classification. Instead of choosing one particular feature selection method, and accepting its outcome as the final subset, different models can be combined using ensemble feature selection approaches. Based on the evidence that there is often not a single optimal feature selection technique, and due to the possible existence of more than one subset of features that discriminates the data equally well, model combination approaches such as boosting, proposed by Freund and Schapire (1997), have been adapted to improve the robustness of final discriminative methods. Similar to the construction of ensemble models for supervised learning, there are two essential steps in creating a feature selection ensemble. The first step involves creating a set of different feature selectors, each providing an output, while the second step aggregates the results of the single models.

3.2.2.1 Ensemble Construction

In ensemble methods for classification, a key point to obtain a good ensemble feature selection is to generate a diverse set of feature selections. There are two efficient ways for this purpose: algorithm perturbation and data perturbation. These two alternatives are also used to achieve diversity in constructing classifier ensembles.

Algorithm perturbation: Using different subsets of features for different classifiers is one of the well known methods for building classifier ensembles. Each ensemble member is associated with its own feature subset. This feature subset is either selected by a certain feature selection algorithm or randomly sampled from the original set of features (Ho (1998)).

In ensemble feature selection, we are interested in the feature selection algorithm perturbation, because we are not only interested to improve classification accuracy, but our objective is also to get a robust feature selection.

Consider a dataset $\mathbf{D}S = (x_i, \ldots, x_m), x_i = (x_i^1, \ldots, x_i^d)$ with m instances and d features. An ensemble of feature selection algorithms $(H_1, \ldots, H_K)$ is applied to DS resulting on K feature subsets $(F_1, \ldots, F_K)$ each one containing n selected features $\mathbf{F}_k = (f_{k,1}, \ldots, f_{k,n})$. For high dimensional data, filters are usually chosen for feature selection,

as they are computationally efficient, fast and independent of the classification algorithm. Diversity and efficiency are also two important keys for creating a successful ensemble (Dietterich (2000)).

Our experimental study will be based on the function perturbation technique where we apply three feature selection algorithms.

Data perturbation: In data perturbation, we deal with ensemble generation through sampling instances. If we are interested in classification only, a classifier would be directly applied on each sample to get a diverse ensemble of classifiers. But our objective here is to obtain a robust feature selection. Thus, we apply a same feature selection algorithm on each sample in order to generate a diverse set of feature selections.

A technique for random sampling of instances is bootstrap (Kohavi (1995)). This technique was discussed in Chapter 1. The basic idea of the statistical bootstrap is sampling with replacement to produce random samples of size m from the original data, each of these is known as a bootstrap sample and each sample is used to provide an estimate of the quantity in question. Bootstrap lays the foundation of two classifier ensemble methods: bagging proposed by Breiman (1996) and random forest, also introduced by Breiman (2001). An ensemble is created by drawing bootstrap samples from the original training data and a feature selection algorithm is applied on each bootstrap sample.

3.2.2.2 First Ensemble Aggregation level : Classifier level

Feature ensemble method based classifiers combination consists in a combination of decisions from multiple classifiers. Each classifier is trained using variations of the feature representation space, obtained by means of feature selection. With this approach, relevant discriminative information contained in features, neglected in a single run of a feature selection method, may be recovered by the application of multiple feature set runs and contribute to the decision through the classifier combination process. While traditional feature selection algorithms try to find the best feature subset which is relevant to both the learning task and the selected inductive learning algorithm, the task of ensemble feature selection by classifiers combination has the main goal of finding a set of feature subsets that promotes disagreement among base classifiers. Opitz (1999) proposed an ensemble feature selection

approach based on GA in order to generate a set of classifiers that are diverse and accurate in their predictions. Tsymbal et al. (2005) introduced a GA based sequential search for ensemble feature selection. Instead of one genetic process, it uses a series of processes, where the goal of each is to build one base classifier. Figure (3.1) illustrates the process of ensemble feature selection based classifiers aggregation.

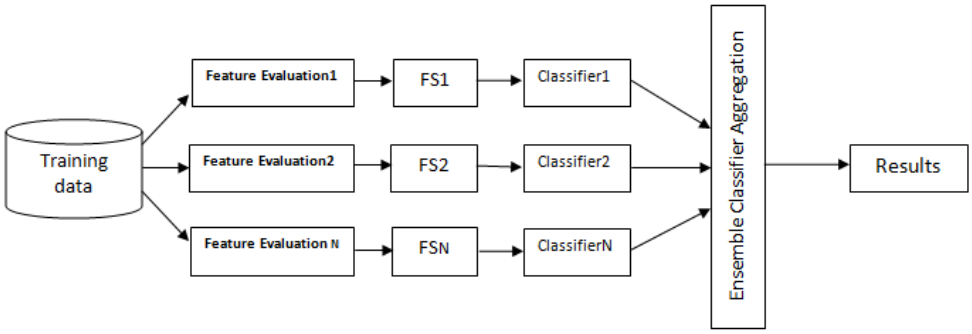


Figure 3.1: Ensemble feature selection based classifier aggregation

In addition to constructing ensembles, the strategy adopted in combining their members is the other fundamental component of any ensemble system. In this section, several rules for combining classifiers are reviewed and grouped into two categories: combination rules for class labels and combination rules for continuous outputs.

Combining class labels: The following aggregation rules are used only if the class labels are available from the classifier outputs. Let $\Omega = \{\omega_1, \omega_2, ..., \omega_C\}$ be the set of classes for T classifiers, $D = \{D_1, D_2, ..., D_T\}$. In the following, we consider that given an instance x to be classified into one of the C classes, the decision of a classifier D_t on class ω_k is represented by

$$d_{t,k} = \begin{cases} 1 & \text{if } D_t \text{ labels } x \text{ in } \omega_k \\ 0 & \text{otherwise.} \end{cases} \tag{3.1}$$

Majority Vote: Majority Vote (MV) chooses the class that receives the largest number of votes among the ensemble classifiers. There are three cases of majority voting (i) unanimous voting when all classifiers agree on the class ω_k; (ii) simple majority when at least more than half the number of classifiers in the ensemble predict the class ω_k, and (iii) plurality voting, or just majority voting, when the class selected is the one that receives the

highest number of votes. Mathematically, majority voting can be written as follows

$$\sum_{t=1}^{T} d_{t,a} = \max_{k=1}^{C} \sum_{t=1}^{T} d_{t,k}, \tag{3.2}$$

where $d_{t,a}$ is the decision of a classifier D_t to choose a class ω_a. For a two class problem, majority voting is an optimal combination method under the conditions that the classifier outputs are independent and that there is an odd number of classifiers.

Weighted Majority Vote: Weighted Majority Vote (WMV) combination rule is proposed by Littlestone and Warmuth (1994). It is useful when certain classifiers are more qualified than others and giving them a higher weight may improve the final decision. The idea of WMV is to assign a weight w_t to each classifier in proportion to its estimated performance. Classifiers' decisions are combined through WMV leading to the choice of class ω_a if

$$\sum_{i=t}^{T} w_t d_{t,a} = \max_{k=1}^{C} \sum_{t=1}^{T} w_t d_{t,k}. \tag{3.3}$$

The selected class is the one receiving the largest total weight. A possible strategy to assign weights is to use the performance of a classifier on a validation dataset, or even on the training dataset. Freund and Schapire (1997) use WMV in Adaboost to combine the ensemble classifiers , it assigns a voting weight to each classifier based on its training error.

Combining continuous outputs: The continuous output is the support given by a classifier to a certain class, and it is usually interpreted as an estimate of its posterior probability when the supports over all classes are normalized to sum up to 1. To introduce the combination rules from the same perspective, Kuncheva et al. (2001) define the decision profile matrix to organize the T classifiers' outputs for a given instance x.
The support given by a classifier D_t to a class ω_k when classifying an example x is denoted $d_{t,k}$. The overall support received by class ω_k is defined using a combination function CF as follows:

$$\mu_k(x) = CF[d_{1,k}(x), ..., d_{T,k}(x)], \tag{3.4}$$

Average: It is the simplest among the algebraic combiners. The support for ω_k is obtained as the average of all classifiers' supports for this class, and is given by:

$$\mu_k(x) = \frac{1}{T}\sum_{t=1}^{T} d_{t,k}(x). \tag{3.5}$$

The class receiving the highest support is selected as the ensemble decision.

Minimum/Maximum Rule: These rules are based on the classifiers outputs which have the minimum or the maximum support. For the maximum rule, the class selected as the ensemble decision is the one receiving the highest support value given by:

$$\mu_k(x) = \max_{t=1}^{T}\{d_{t,k}(x)\}. \tag{3.6}$$

The minimum rule selects the class having the minimum support among the classifiers outputs, it is defined as:

$$\mu_k(x) = \min_{t=1}^{T}\{d_{t,k}(x)\}. \tag{3.7}$$

Product Rule: Product rule assumes reliable support estimates, it is adequate only if the single classifiers are independent. Such classifiers may be formed by training on different feature sets. For this combination method, the supports obtained by the classifiers are multiplied. However, it is very sensitive to supports that are very small and close to zero. The product rule is defined as

$$\mu_k(x) = \frac{1}{T}\prod_{t=1}^{T} d_{t,k}(x) \tag{3.8}$$

The sum rule: This rule assumes also independent classifiers with small differences in their outputs.The errors of the supports provided by such an ensemble of classifiers are averaged by the summation. A good example may be an ensemble of classifiers based on the same learning algorithm in the same feature space, but trained with different training sets like in Bagging where the successive training sets are bootstrap replicates from the original training data.

3.2.2.3 Second Ensemble Aggregation level : Feature Selector level

The concept of ensemble feature selection based feature selectors aggregation was recently introduced by Saeys et al. (2008). Ensemble feature selection techniques use an idea similar to ensemble learning for classification (Dietterich (2000)). In a first step, a number of

different feature selectors are used, and in a final phase the output of these separate selectors is aggregated and returned as the final ensemble result. Similar to the case of supervised learning, ensemble techniques might be used to improve the robustness of feature selection techniques. Different feature selection algorithms may yield feature subsets that can be considered local optima in the space of feature subsets, and ensemble feature selection might give a better approximation to the optimal subset or ranking of features. Also, the representational power of a particular feature selector might constrain its search space such that optimal subsets cannot be reached. Ensemble feature selection could help in alleviating this problem by aggregating the outputs of several feature selectors. This concept was specially applied for high dimensional data with few samples as discussed by Saeys et al. (2008) and Schowe and Morik (2011). Ensemble concept for feature selection can be also in the form of parallel application of multiple feature algorithms. Mitchell et al. (2014) proposed a parallel implementation of the bootstrap resampling step and combination of results of rank product method for feature selection for the identication of differentially expressed genes. Figure (3.2) illustrates the process of ensemble feature selection based selector aggregation.

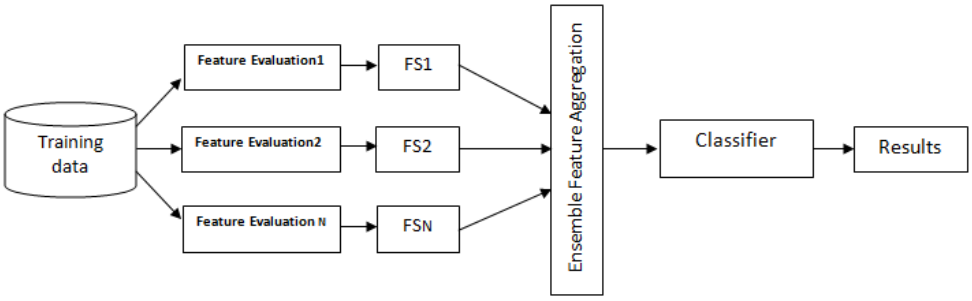


Figure 3.2: Ensemble feature selection based selectors aggregation

The most important decision in this level of ensemble feature subsets aggregation is how to combine the resulting feature lists from the multiple algorithms into a single decision for each feature. There exist simple aggregation techniques and other more complicated ones. Often, these techniques are used to aggregate either feature weights or ranks. In the following, we introduce some of these aggregation techniques.

Aggregating feature weights

Weighted Mean Aggregation: The Weighted Mean Aggregation (WMA) method uses the weights of all the features obtained by the different selected subsets then for each feature the weights mean is calculated. To select the final set of features for a signature of size s, the s features with the highest weighted mean are selected (Abeel et al. (2010)).

Feature weight-rank based aggregation technique : We propose the feature weight-rank based aggregation technique (WRA) which uses the weights of all the features obtained by the different selected subsets then for each feature the total weight is calculated. Also the complete ranking of all the features is used to sum the ranks over all ranking lists for each feature. The total weight is then divided by the total rank. To select the final set of features for a signature of size s, the s features with the highest scores are selected.

Aggregating feature ranks

Complete linear aggregation: The complete linear aggregation (CLA) method uses the complete ranking of all the features then the ranks over all ranking lists are summed for each feature. To select the final set of features for a signature of size s, the s features with the lowest summed rank are selected (Abeel et al. (2010)).

Robust RankAggregate: The Robust RankAggregate (RRA) method, proposed by Kolde et al. (2012), detects features that are ranked consistently better than expected under the null hypothesis of uncorrelated inputs and assigns a significance score for each feature. The underlying probabilistic model makes the algorithm parameter free and robust to outliers, noise and errors. Significance scores also provide a rigorous way to keep only the statistically relevant features in the final list. These properties make this approach robust and compelling for many settings. For each item, the algorithm looks at how the item is positioned in the ranked lists and compares this to the baseline case where all the preference lists are randomly shuffled. As a result, a p-value is assigned for all items, showing how good it is positioned in the ranked lists than what is expected by chance. This P-value is used both for re-ranking the items and deciding their significance.

GA based aggregation: The aim of rank aggregation when dealing with feature selection is to find the best list, which would be the closest as possible to all individual ordered lists all together. Pihur et al. (2009) proposed the GA based aggregation (GAA) which treats rank aggregation as an optimization problem. By looking at $argmin(D, \sigma)$, $argmin$ gives a list σ at which the distance D with a randomly selected ordered list is minimized. In this optimization framework the objective function is given by :

$$F(\sigma) = \sum_{k=1}^{K} w_k \times D(\sigma, L_k), \tag{3.9}$$

where w_k represent the weights associated with the lists L_k, D is a distance function measuring the distance between a pair of ordered lists and L_k is the k^{th} ordered list of cardinality n. The best solution to look for σ^* which would minimize the total distance between σ^* and L_k is given by:

$$\sigma^* = argmin \sum_{k=1}^{K} w_k \times D(\sigma, L_k). \tag{3.10}$$

Measuring the distance between two ranking lists is classical and several well-studied metrics are discussed by Carterette (2009); Kumar and Vassilvitskii (2010), including the Kendall's tau distance and the Spearman footrule distance. Spearman footrule distance between two given rankings lists L and σ is defined as the sum overall the absolute differences between the ranks of all unique elements from both ordered lists combined. The Kendall's tau distance between two ordered rank list L and σ is given by the number of pairwise adjacent transpositions needed to transform one list into another (Dinu and Manea (2006)). This distance can be seen as the number of pairwise disagreements between the two rankings. The introduced optimization problem is a typical integer programming problem. The presented method uses GA for rank aggregation.

Other aggregation techniques

Feature occurrence frequency: The feature occurrence frequency based aggregation (OFA) obtains the final feature selection by calculating the number of occurrences of each feature over all lists and ranking them based on their occurrence frequency. This ranking

technique favors features appearing in the maximum number of feature subsets built based on function or data perturbation.

Common features: This aggregation technique uses a simple process which selects features shared by all feature selectors.

3.2.3 Comparative study

In this section we address two issues involving high dimensional data. The first issue explores the behavior of ensemble method feature aggregation when analyzing data with hundreds or thousands of dimensions in small sample size situations. The second issue deals with huge data set with a massive number of instances and where feature selection is used to extract meaningful rules from the available data.

For the ensemble construction step, we apply function perturbation where we select a signature of a given size s of best features from each of the three feature ranking lists obtained as output. Filter methods give as output all the input features ranked according to their score so we don't have any indication about the feature set size required to have a good classification performance. A way to approximate the best solution would be to evaluate many feature set cardinalities with a classification algorithm and to keep the cardinality that gives the best classification performance.

3.2.3.1 Datasets

The experiments for the first case were conducted on CNS data set, a large data set concerned with the prediction of central nervous system embryonal tumor outcome based on gene expression (Pomeroy et al. (2002)). This data set includes 60 samples containing 39 medulloblastoma survivors and 21 treatment failures. These samples are described by 7129 genes. We consider also the Leukemia microarry gene expression dataset introduced by Golub et al. (1999). It consists of 72 samples which are all acute leukemia patients, either acute lymphoblastic leukemia (47 ALL) or acute myelogenous leukemia (25 AML). The total number of genes to be tested is 7129.

For the second case a credit dataset is used. The credit dataset covers a sample of 2970 instances of credit consumers where 2523 instances are creditworthy and 446 are not. Each

credit applicant is described by a binary target variable and a set of 22 input variables where 11 features are numerical and 11 are categorical. Table (3.1) displays the characteristics of the datasets that have been used for evaluation.

Table 3.1: Datasets summary

Names	Credit	CNS	Leukemia
Total instances	2970	60	72
Total features	22	7129	7129
Number of classes	2	2	2
Missing Values	Yes	No	No

3.2.3.2 Feature selection algorithms

Our feature selection ensemble is composed by three different filter selection algorithms, Relief algorithm (Kira and Rendell (1992)), CBFS (Hall (2000)) and IG (Quinlan (1993)).

The aggregation of these filters in the feature selection level is performed by three aggregation techniques described. The first is choosing the selected features shared by the three methods (Common), the second is the WMA and finally the GAA method with its two distance alternatives: Kendall and Spearman.

3.2.3.3 Classifiers

We trained our approach using two well-known data mining algorithms, namely SVM and kNN. These algorithms and feature selection algorithms are available in Weka 3.7.0 machine learning package (Bouckaert et al. (2009)).

The aggregation in the classifiers level is performed based on five well known combination rules described above namely, the majority vote, the average probability, the product probability, the minimum probability and the maximum probability combination rule.

3.2.3.4 Performance metrics

To evaluate the classification performance of each setting and perform comparisons, we used several characteristics of classification performance all derived from the confusion

matrix (Costa et al. (2007)). These metrics were detailed in Chapter 1. We redefine briefly these evaluation metrics.

The precision is the percentage of positive predictions that are correct. The Recall (or sensitivity) is the percentage of positive labeled instances that were predicted as positive. The F-measure can be interpreted as a weighted average of the precision and recall. It reaches its best value at 1 and worst score at 0.

A ROC curve is a plot of the sensitivity (or the TP rate) against one minus its specificity (or the FP rate), as the cut-off criterion for indicating a positive test is varied. This plot depicts relative trade-offs between TPs and false positives. We use the area under the curve (ROC Area) as another performance metric.

3.2.3.5 Performance analysis

We consider information retrieval measures of datasets when individual filter methods are applied, using the learning algorithms by 10-fold CV. Then, we apply the ensemble feature selection, the first is based on the classifier aggregation denoted by ECA, then based on the feature set aggregation, denoted by ESA. We measured the performance of those methods. Tables (3.2) - (3.4) show the results of the experiments.

For Credit data set, ensemble methods on both aggregation levels give results equal or worse than baseline algorithms. We see that with SVM classifier, IG gives the best performance. Moreover, CBFS gives the best performance with kNN classifier.

For CNS data set, IG with SVM classifier gives the best individual performance. Only one ensemble classifier based aggregation method gives an equal performance, using MV, while other methods yield worse results than the best baseline algorithm. However, with selector aggregation methods, all techniques improve classification compared to baseline algorithms with best performance obtained with Common features rule followed by GAA (Spearman). The same conclusions are made with kNN classifier, except for MV rule which in this setting deteriorates accuracy compared to baseline algorithms.

For the Leukemia data set, Relief is the best individual feature selection algorithm with both SVM and kNN classifiers. MV gives a performance equal to Relief with SVM, while other classifier based aggregation methods give slightly smaller results. Selector based aggregation techniques have similar behaviour, except for GAA (Kendall) which outperforms

Table 3.2: Performance results summary for the Credit dataset

SVM					
	Algorithm	Precision	Recall	F-Measure	ROC Area
Baseline	CBFS	0.851	0.994	0.917	0.505
	Relief	0.85	1	0.919	0.5
	IG	0.868	0.907	0.887	**0.563**
ECA	Majority V	0.851	0.994	0.917	0.505
	Average	0.851	0.994	0.917	0.505
	Product	0.851	1	0.919	0.505
	Max	0.85	1	0.919	0.505
	Min	0.851	1	0.919	0.505
ESA	Common	0.85	1	0.919	0.5
	WMA	0.769	0.847	0.785	0.5
	GAA(Kendall)	0.85	1	0.919	0.5
	GAA(Spearman)	0.851	0.994	0.917	0.505
kNN					
	Algorithm	Precision	Recall	F-Measure	ROC Area
Baseline	CBFS	0.864	0.959	0.909	**0.675**
	Relief	0.862	0.932	0.895	0.602
	IG	0.86	0.94	0.898	0.607
ECA	Majority V	0.86	0.967	0.91	0.539
	Average	0.86	0.957	0.906	0.67
	Product	0.86	0.937	0.897	0.658
	Max	0.86	0.937	0.897	0.67
	Min	0.86	0.937	0.897	0.643
ESA	Common	0.861	0.931	0.895	0.596
	WMA	0.864	0.938	0.899	0.644
	GAA(Kendall)	0.863	0.938	0.899	0.63
	GAA(Spearman)	0.866	0.941	0.902	0.645

Table 3.3: Performance results summary for the CNS dataset

SVM					
	Algorithm	Precision	Recall	F-Measure	ROC Area
Baseline	CBFS	0.700	0.718	0.709	0.573
	Relief	0.632	0.615	0.623	0.474
	IG	0.737	0.718	0.727	**0.621**
ECA	Majority V	0.737	0.718	0.727	**0.621**
	Average	0.737	0.718	0.727	0.58
	Product	0.737	0.718	0.727	0.58
	Max	0.704	0.487	0.576	0.542
	Min	0.704	0.760	0.731	0.553
ESA	Common	0.9	0.923	0.911	**0.866**
	WMA	0.825	0.846	0.835	0.756
	GAA(Kendall)	0.805	0.846	0.825	0.733
	GAA(Spearman)	0.875	0.897	0.886	0.83
kNN					
	ALgorithm	Precision	Recall	F-Measure	ROC Area
Baseline	CBFS	0.677	0.538	0.600	0.531
	Relief	0.659	0.692	0.675	0.513
	IG	0.727	0.615	0.667	**0.593**
ECA	Majority V	0.688	0.564	0.62	0.544
	Average	0.688	0.564	0.62	0.563
	Product	0.688	0.564	0.62	0.571
	Max	0.739	0.436	0.548	0.574
	Min	0.739	0.436	0.548	0.574
ESA	Common	0.878	0.923	0.9	**0.842**
	WMA	0.837	0.923	0.878	0.795
	GAA(Kendall)	0.787	0.949	0.86	0.736
	GAA(Spearman)	0.841	0.949	0.892	0.808

Table 3.4: Performance results summary for the Leukemia dataset

SVM					
	Algorithm	Precision	Recall	F-Measure	ROC Area
Baseline	CBFS	0.958	0.979	0.968	0.949
	Relief	0.979	0.979	0.979	**0.969**
	IG	0.938	0.957	0.947	0.919
ECA	Majority	0.979	0.979	0.979	**0.969**
	Average	0.979	0.979	0.979	0.968
	Product	0.978	0.978	0.978	0.959
	Max	0.92	0.979	0.948	0.966
	Min	0.978	0.978	0.978	0.959
ESA	Common	0.958	0.979	0.968	0.949
	WMA	0.972	0.972	0.972	0.969
	GAA(Kendall)	0.986	0.986	0.986	**0.98**
	GAA(Spearman)	0.972	0.972	0.972	0.969
kNN					
	Algorithm	Precision	Recall	F-Measure	ROC Area
Baseline	CBFS	0.938	0.957	0.947	0.911
	Relief	0.957	0.957	0.957	**0.936**
	IG	0.956	0.915	0.935	0.92
ECA	Majority V	0.957	0.957	0.957	0.939
	Average	0.957	0.957	0.957	**0.958**
	Product	0.957	0.957	0.957	**0.958**
	Max	0.92	0.979	0.948	0.956
	Min	0.92	0.979	0.948	0.956
ESA	Common	0.978	0.936	0.957	0.947
	WMA	0.973	0.972	0.972	**0.951**
	GAA(Kendall)	0.973	0.972	0.972	**0.951**
	GAA(Spearman)	0.958	0.958	0.958	0.938

all individual and ensemble settings. With kNN classifier, all ensemble method settings with their two aggregation levels improve individual performances. Average and Product rules give the best results for the classifier based aggregation level. WMA and GAA (Kendall) give the best performances in the selector based aggregation level.

Based on the analysis above, we conclude that if the data set size is very small and the number of features exceeds the number of instances ensemble methods can be efficient in improving the classification accuracy. This improvement is especially noticeable if we introduce aggregation in the pre-processing step before the learning process. In case of big data set in terms of instances where their number exceeds the number of features, ensemble methods are not useful to improve classification accuracy neither with the classifier based aggregation nor with selector based aggregation level.

3.2.4 Discussion

In this section, we apply three different feature selection methods on three data sets resulting in three SFSs for each dataset. Then in a first setting, we apply a classification algorithm on the projection of each feature subset on the training data. We then aggregate the classification results of the ensemble. In a second setting, the three SFSs obtained initially are combined in order to obtain a final individual feature subset before proceeding to the classification step. The comparison of the two settings performances yields the following conclusions.

On most cases where we have high dimensional data and small size of samples, the ensemble results, obtained by one ensemble aggregation level or the other, outperform those obtained by the application of a single feature selection algorithm followed by a single classifier. For this kind of data sets, the best performance results are thus achieved even by classifiers or selectors aggregation, with special high values when feature selectors aggregation is outperforming.

For data set with small dimensionality and large samples, the best performance results are obtained by applying only a baseline feature selection algorithm and ensemble methods are not efficient.

A possible explanation of the performance of feature selection aggregation on high dimensional data sets, and not on data sets with small dimensionality, is that on the latter

individual feature subsets obtained by different feature selection methods may be very similar as the initial number of features is small. However, in the case of high dimensional data sets, obtained feature subsets from the ensemble feature selection process may be very different as the feature space is very large. Thus the features combination effect on classification performance will be much more apparent in case of high dimensional data sets.

Therefore, sample size may indicate if it is advantageous to apply ensemble methods or not when classification accuracy is the considered performance criterion. Stability is another important criterion for evaluating feature selection results and in terms of this performance metric we expect that feature selector based aggregation ensembles will be preferred as they focus on improving classification results by strengthening feature selection results and working with a single aggregated feature subset. It is not the case for classifier ensembles which focus on strengthening classification results without a special care to feature selection phase and working with many feature subsets.

Next, we focus on enhancing feature selection stability by proposing a new robust ensemble feature selector aggregation methods.

3.3 Robust ensemble feature selection based on multiple classifiers performance

Classifier ensembles are efficient in enhancing accuracy, however they bring more features to be considered than any single classifier. In case of genomic applications like cancer diagnosis, biologists would not appreciate having a high number of features. However, machine learning researchers would defend ensembles due to their superior classification performance. Hence, it is important to find a trade-off. Different feature selection algorithms are built based on optimizing different relevance criteria. Thus, they will have different biases and may produce different results. Okun (2011) mentioned that despite such a difference, if the same gene appears in multiple SFSs obtained by different algorithms, and produce accurate classifiers, it is indeed important. We propose a robust feature selection aggregation technique based on this idea.

The proposed ensemble feature selection framework consists of two steps. The first is the ensemble creation and the second is the ensemble outputs aggregation. For the first step, we experiment two alternatives and compare their performances. The first is function perturbation where we use different feature selection algorithms and conduct them on

the original data. The second one uses data perturbation to construct a feature selectors ensemble where we sample the data and use the same feature selection method for each sample. To combine their results, we propose a robust aggregation technique and compare it to existing ones. These two important steps of the proposed method are detailed in the following.

3.3.1 Ensemble Construction

Like in supervised learning, the generation of a set of diverse component learners is one of the keys to the success of ensemble learning. Variation in the feature selectors can be achieved by various methods, such as data perturbation and function perturbation. Data perturbation tries to run component learners with different sample subsets. Function perturbation refers to those ensemble feature selection methods in which the component learners are different from each other. The basic idea is to leverage on the strengths of different algorithms to obtain robust feature subsets. Existing ensemble feature selection methods in this category differ mainly in their aggregation procedure.

3.3.1.1 Algorithm perturbation

For high dimensional data, filters are usually chosen as long as they are computationally efficient, fast and independent of the classification algorithm. Dietterich (2000) demonstrated that diversity and efficiency are also two important keys for creating a successful ensemble. Thus to create the selectors ensemble, we choose three popular and successful filters which are based on different selection criteria. These algorithms are t-test, mRMR (Peng et al. (2005)) and Relief (Kira and Rendell (1992)).

After the application of the three feature selectors, we select a feature subset of best features from each of the three feature ranking lists obtained as output. Filter methods give as output all the input features ranked according to their score so we don't have any indication about the feature set size required to have a good classification performance. A way to approximate the best solution would be to evaluate many feature set cardinalities with a classification algorithm and to keep the cardinality that gives the best classification performance.

3.3.1.2 Data perturbation

Starting from a particular training set, our aim is to generate a diverse set of feature selections. To generate diversity in the selection, the feature selection method is run on different training sub-samples. To this end, we make use of the bootstrapping method, a well-established technique in statistics to reduce variance. By drawing different bootstrap samples with replacement from the training data, we can apply a filter to each of these bootstrap samples and thus obtain a diverse set of feature rankings.

3.3.2 Ensemble feature selector aggregation based on multiple classifiers performance

The choice of the technique to use for the aggregation step is an important decision for ensemble feature selection. We propose robust feature aggregation techniques to combine the results of the different feature subsets obtained either by function or data perturbation as described above. To this end, we propose to take advantage of both classifier ensembles and feature selection ensembles benefits in order to improve classification but also feature selection stability.

In fact, we have seen that classifier ensembles are often efficient for enhancing classification performance as they combine the decisions of many classifiers by voting or averaging. However, they bring more features to consideration as classifiers are trained using several feature subsets. This is not an issue for machine learning researchers but biologists would like to have a single feature subset that is efficient and stable in the same time. So, to reconcile interests of two groups of researchers, we propose an ensemble aggregation technique that uses classification performance of multiple classifiers trained on different feature subsets to guide the selection of features corresponding to high accuracies. We use this concept to propose a robust aggregation technique.

Chan et al. (2008) proposed a classification accuracy based aggregation (CAA) that assigns a score to each feature in the different lists as the sum of accuracies for all classifiers that include that feature. This score is given by:

$$Sc(f_j) = \sum_{k=1}^{K} e_{kj} * (1 - \beta_k) \tag{3.11}$$

where β_k is the normalized error of the k^{th} classifier trained on the projection of the k^{th} feature subset on the data, K is the number of classifiers (lists) in the ensemble and $e_{kj} = 1$ if f_j is a feature selected for the k^{th} classifier and zero otherwise. Such a scoring scheme favors the features that lead to more accurate classification but it is considered simple. We propose a sophisticated and robust aggregation method to optimize classification accuracy and stability of feature selection based on features reliability assessment. The proposed approach is detailed in the following.

Reliability Assessment Aggregation

The reliability assessment aggregation method (RAA) is based on measuring feature selection algorithm's confidence and their conflict with other selection algorithms in order to assign a reliability factor guiding the final feature selection. The confidence and conflict are calculated based on weights of selected features and according to the classification performance of a classifier obtained with the projection of a selected feature set.

The opinions given by the ensemble of feature selection algorithms are represented as weights given to each selected feature. To enhance robustness of the final selection, these opinions are associated with a confidence level presenting the belief on the feature selection decision. The RAA approach determines the conflict level of each feature selection algorithm by measuring the similarity between its opinion and confidence, and those of the other algorithms in the ensemble. Based on those conflict levels, a reliability rate is associated to each algorithm, such as a reliable algorithm is the one which is confident and non-conflicting at the same time (Garcia and Puig (2003)). The final decision is obtained by multiplying the reliability factors by the original selection algorithm opinions. Our robust aggregation technique involves two steps. The first one is the features' confidence calculation based on their weights and their associated classification error. The second one is the reliability assessment and decision making.

Confidence Calculation We note that the trained feature selectors ensemble resulted in K feature subsets. A classifier is trained on each newly obtained training set containing only the feature subset obtained by each feature selector. The overall accuracies of the K classifiers by 10-fold CV are determined. Each classifier is used here to evaluate an individual feature subset and assigns a confidence level according to the classification performance

obtained with the projection of that feature subset. Any classification algorithm could be used but it is preferable to choose a simple classifier as we are still in a preprocessing phase. For this purpose, we use a kNN classifier.

The K individual feature subsets are then merged into a single feature set containing all selected features. Let $FS = (f_1, \ldots, f_S)$ be the resulting merged feature set and $op_{k,j}$ denotes the opinion of the k^{th} feature set algorithm H_k about the selected feature f_j. This opinion is the weight assigned by H_k to feature f_j and it is equal to zero if feature f_j is not selected by H_k.

A confidence level $conf_{k,j}$ is assigned to each selection algorithm H_k about each opinion $op_{k,j}$. The confidence is a weight calculated as follow:

$$conf_{k,j} = op_{k,j} * \log(\frac{1}{\beta_k}), \tag{3.12}$$

where β_k is the normalized error of the kNN classifier trained on the projection of the k^{th} feature subset on the data. Confidences are then normalized.

Reliability Assessment and Decision Making Given the opinions of K feature selection algorithms about the selection of a feature f_j, $Op_j = \{op_{k,j}, k = 1, \ldots, K\}$, and given the confidences associated with those opinions, $Conf_j = \{conf_{k,j}, k = 1, \ldots, K\}$, the conflict of each selection algorithm is formulated, by first measuring the similarity between its opinions and those of the other algorithms in the ensemble, as follows:

$$Sim_k(Op_j) = 1 - \frac{1}{(K-1)} \sum_{t=1, t\neq k}^{K} \mid op_{k,j} - op_{t,j} \mid . \tag{3.13}$$

Then, algorithm's confidences similarity with the rest of confidences, $Sim_k(Conf_j)$, is calculated the same way as in Eq. (3.13). Based on these calculations, the conflict raised by an algorithm is defined as

$$Conflict_{k,j} = Sim_k(Conf_j)[1 - Sim_k(Op_j))]. \tag{3.14}$$

Conflicting selectors are those with similar confidences to the agreeing selectors but completely different opinions from theirs. The conflict measure will affect selection algorithm's reliability for a feature f_j which is calculated as follows

$$rel_{k,j} = conf_{k,j}(1 - Conflict_{k,j}). \tag{3.15}$$

Finally, the original opinions about the features are adjusted by multiplying them by the associated reliability factors after being normalized. The selected features are the best ranked ones according to their adjusted opinion. The robust aggregation method is implemented using matlab software.

3.3.3 Experimental study

In this section, we compare the performance of our proposed ensemble feature selection method and those of other methods. Our experimental data consists of seven cancer diagnosis microarrays data sets described in Chapter 2.

3.3.3.1 Performance metrics

We use 10-fold stratified CV to evaluate results of the different ensemble feature selection methods based on the classification performance of SVM classifier on the seven data sets. The MCE of a classifier is defined as the proportion of misclassified instances over all classified instances. This metric is important and always used to evaluate feature selection algorithms for classification tasks. However, it is not sufficient given that there is no best way to evaluate any system and different metrics give different insights into how a feature selection algorithm performs. Hence, as done before, we evaluate also the stability to compare our proposed method to other existing ensemble feature selection methods.

3.3.3.2 Results analysis

We report here the experimental evaluations on the seven cancer diagnosis microarray data sets considered. The classification performance of our proposed method (RAA) is compared to several ensemble aggregation techniques discussed before. Results on the stability for the different aggregation schemes are also discussed.

We report also the classification performance of ensemble classifier aggregation referred to as ECA, which instead of combining selected feature subsets (SFS), it aggregates decisions of classifiers built on each individual SFS. The aggregation technique used for

ECA is the simple and efficient majority vote aggregation method that aggregates class labels. Note that ECA has not a corresponding stability performance, as it is built using several feature subsets and not a single one. Its unique objective is to enhance predictive performance.

We compare the proposed and existing feature selection ensemble methods built by perturbing the baseline algorithm to the data perturbation setting, and analyze the stability and classification performance for each of the seven cancer data sets. To have a clearer analysis of the classification performance, we underlined the three best results for each data set.

Tables (3.5) - (3.8) show the MCE and stability (Stab) results obtained from all settings for the seven data sets. The MCE and stability values reported are obtained by averaging 10 MCE and stability results. These results are obtained by varying the feature subset size with 10 SFS cardinalities ranging from 10 to 100 features. All detailed MCE results for all SFS cardinalities are reported in Appendix A. Stability curves of all data sets are also given in Appendix A.

3.3.3.3 Data perturbation

Data perturbation with Relief

Table (3.5) shows classification and stability results of the data perturbation setting using Relief as a baseline algorithm. In terms of classification performance, ensemble methods improve the baseline performance for most cases. A special high classification performance is noticed with ECA for all data sets. WMA is performing well for five out of seven data sets. RAA and OFA follow in the third place.

In terms of stability of feature selection, results show that RAA and WMA improve stability comparing to the baseline algorithm and give the best results for the the data perturbation setting using Relief for all data sets. RRA give also good results. It is noticed that CLA gives poor stability results. This technique relies on feature ranking. Therefore, RAA and WMA are efficient with this setting, both in terms of classification performance and stability of feature selection.

Table 3.5: Classification error rates and stability of ensemble methods with Relief and the data perturbation setting.

Dataset		Relief	ECA	RAA	WMA	CLA	CAA	RRA	OFA
DLBCL	MCE	0.108	0.061	0.079	0.074	0.091	0.127	0.104	0.098
	Stab	0.600	-	0.848	0.855	0.243	0.764	0.785	0.599
Bladder	MCE	0.239	0.168	0.193	0.187	0.222	0.274	0.219	0.171
	Stab	0.454	-	0.634	0.665	0.082	0.562	0.585	0.381
Lymph	MCE	0.142	0.078	0.08	0.084	0.106	0.215	0.1	0.157
	Stab	0.498	-	0.732	0.748	0.095	0.641	0.651	0.385
Prostate	MCE	0.254	0.095	0.129	0.087	0.132	0.178	0.137	0.175
	Stab	0.577	-	0.854	0.851	0.297	0.697	0.807	0.618
Breast	MCE	0.432	0.435	0.464	0.45	0.469	0.487	0.436	0.448
	Stab	0.569	-	0.589	0.574	0.065	0.554	0.347	0.414
CNS	MCE	0.428	0.349	0.405	0.33	0.41	0.425	0.433	0.373
	Stab	0.478	-	0.806	0.819	0.264	0.622	0.756	0.629
Lung	MCE	0.020	0.015	0.021	0.018	0.023	0.034	0.022	0.01
	Stab	0.799	-	0.902	0.898	0.366	0.833	0.846	0.681

Data perturbation with mRMR

Table (3.6) shows that classification results of mRMR are among the best results for DLBCL, Bladder and CNS data sets. RAA and ECA are competitive, both of them achieve minimum MCE for five data sets. WMA followed by OFA have also good classification performances in some data sets.

Stability results with mRMR and data perturbation setting show that OFA is the most stable. CAA and mRMR have similar stability. RAA and WMA are following with smaller stability results and CLA gives poor stability as noticed before. In general, best stability results obtained by this setting are worst than those obtained by Relief based ensemble methods, specially for Breast and CNS data sets. We can deduce that RAA, WMA and OFA are the ensemble methods achieving the best trade-off between classification performance and stability for this setting. If we focus in classification results, RAA and WMA are favored and vice versa if stability is more important as evaluation metric.

Data perturbation with t-test

Table 3.6: Classification error rates and stability of ensemble methods with mRMR and the data perturbation setting.

Dataset		mRMR	ECA	RAA	WMA	CLA	CAA	RRA	OFA
DLBCL	MCE	0.087	0.048	0.088	0.097	0.096	0.166	0.114	0.116
	Stab	0.562	-	0.548	0.529	0.048	0.566	0.295	0.580
Bladder	MCE	0.132	0.158	0.138	0.155	0.135	0.161	0.187	0.187
	Stab	0.514	-	0.469	0.450	0.043	0.526	0.304	0.564
Lymph	MCE	0.044	0.033	0.042	0.027	0.049	0.14	0.053	0.049
	Stab	0.631	-	0.595	0.583	0.035	0.629	0.429	0.653
Prostate	MCE	0.135	0.096	0.122	0.107	0.122	0.225	0.125	0.113
	Stab	0.725	-	0.678	0.679	0.050	0.707	0.528	0.729
Breast	MCE	0.328	0.305	0.319	0.296	0.337	0.35	0.328	0.317
	Stab	0.390	-	0.336	0.309	0.022	0.365	0.120	0.412
CNS	MCE	0.35	0.42	0.403	0.416	0.426	0.441	0.401	0.435
	Stab	0.378	-	0.328	0.322	0.015	0.330	0.119	0.410
Lung	MCE	0.021	0.010	0.016	0.015	0.019	0.024	0.017	0.009
	Stab	0.807	-	0.678	0.670	0.103	0.748	0.654	0.802

The results of the data perturbation setting using t-test are reported in Table (3.7). In terms of classification performance, ensemble methods are efficient and outperform the baseline algorithm in most cases. For this setting also, RAA is competing with ECA. WMA has also good performance for four data sets.

Stability results show that t-test is the most stable, but here also these good stability results are coupled with poor classification results. Thus, t-test can not be considered as reliable if both evaluation criteria are considered. We notice also high stability results for CAA and OFA. RAA follows with slightly smaller performance. CLA, the ranking based ensemble method have the same behaviour as for other data perturbation settings. Stability results corresponding to CNS data set are specially poor for all algorithms. For Breast cancer data set, all algorithms have the same and perfect stability, that is however coupled with a high MCE (0.515%). In general, for this setting also, RAA proves its efficiency both for classification and stability performances in many cases.

Table 3.7: Classification error rates and stability of ensemble methods with t-test and the data perturbation setting.

Dataset		t-test	ECA	RAA	WMA	CLA	CAA	RRA	OFA
DLBCL	MCE	0.182	0.078	0.166	0.178	0.178	0.166	0.166	0.182
	Stab	0.829	-	0.735	0.677	0.037	0.768	0.549	0.780
Bladder	MCE	0.071	0.103	0.087	0.1	0.103	0.197	0.148	0.109
	Stab	0.845	-	0.748	0.700	0.027	0.769	0.372	0.784
Lymph	MCE	0.058	0.049	0.064	0.055	0.058	0.28	0.06	0.066
	Stab	0.782	-	0.642	0.567	0.021	0.693	0.421	0.699
Prostate	MCE	0.119	0.08	0.12	0.114	0.134	0.211	0.115	0.123
	Stab	0.797	-	0.721	0.687	0.033	0.746	0.562	0.747
Breast	MCE	0.526	0.515	0.515	0.515	0.515	0.515	0.515	0.515
	Stab	1	-	1	1	1	1	1	1
CNS	MCE	0.468	0.363	0.358	0.368	0.366	0.43	0.4	0.383
	Stab	0.430	-	0.315	0.268	0.012	0.329	0.080	0.354
Lung	MCE	0.041	0.015	0.022	0.032	0.027	0.036	0.021	0.028
	Stab	0.894	-	0.781	0.742	0.085	0.810	0.735	0.838

3.3.3.4 Algorithm perturbation

Table (3.8) shows classification results of the algorithm perturbation setting. For two data sets (Prostate and Breast), mRMR gives the optimal MCE. It gives also a good performance for CNS data set. ECA is a good choice for four data sets, followed by RRA and WMA. RAA, our proposed method, is not specially efficient with algorithm perturbation setting.

Stability performance of ensemble methods are smaller than t-test baseline method for all data sets. However, this is not enough for t-test to be considered as good choice, as it gives poor or modest results in terms of classification performance. OFA gives the best stability results for the ensemble methods.

Table 3.8: Classification error rates and stability of ensemble methods with the algorithm perturbation setting.

Dataset		Relief	mRMR	t-test	ECA	RAA	WMA	CLA	CAA	RRA	OFA
DLBCL	MCE	0.081	0.105	0.209	0.084	0.098	0.119	0.101	0.082	0.084	0.113
	Stab	0.578	0.451	0.798	-	0.542	0.425	0.389	0.554	0.437	0.651
Bladder	MCE	0.142	0.152	0.145	0.132	0.097	0.097	0.132	0.113	0.103	0.119
	Stab	0.398	0.404	0.786	-	0.436	0.355	0.407	0.376	0.418	0.585
Lymph	MCE	0.084	0.04	0.049	0.033	0.035	0.04	0.018	0.082	0.033	0.033
	Stab	0.401	0.557	0.69	-	0.534	0.395	0.290	0.587	0.437	0.598
Prostate	MCE	0.162	0.084	0.114	0.09	0.11	0.103	0.107	0.157	0.142	0.127
	Stab	0.579	0.659	0.728	-	0.587	0.53	0.522	0.631	0.599	0.653
Breast	MCE	0.439	0.334	0.515	0.395	0.515	0.355	0.439	0.493	0.493	0.497
	Stab	0.263	0.256	1	-	1	0.263	0.450	0.159	0.450	0.626
CNS	MCE	0.446	0.4	0.445	0.423	0.383	0.401	0.421	0.481	0.47	0.38
	Stab	0.442	0.247	0.318	-	0.255	0.217	0.261	0.244	0.264	0.366
Lung	MCE	0.019	0.013	0.031	0.005	0.015	0.011	0.013	0.010	0.009	0.004
	Stab	0.762	0.748	0.852	-	0.614	0.525	0.629	0.639	0.673	0.726

3.3.4 Discussion

The highest stability results for all data sets are obtained by our proposed method RAA and by WMA using data perturbation with Relief baseline algorithm. ECA gives often excellent classification performances. However, this is not good enough, since there is not a corresponding stability performance. In fact, the objective of ECA is not to have a stable feature selection but to enhance predictive performance by aggregating classifier results built on different feature subsets. Therefore, if the interest is in classification performance, ECA is the technique to use to get the best results. However, if we search for techniques to achieve good classification and feature selection stability at the same time, RAA, our proposed method based on conflict resolution and reliability assessment, is the best solution. WMA, which is a simple technique that aggregates feature weights, has also proved its efficiency. OFA is often the most stable ensemble feature selection method. Finally experiments show that CLA, which combines feature ranks, is not efficient specially in terms of stability performance.

3.4 Conclusion

In this chapter, we first proceeded to a comparative study between different levels of ensemble feature selection, classifier level aggregation and selector based aggregation. We also studied the effect of sample size data on the classification. Our objective was to study the characteristics and to compare the performance of each setting but especially to search for the level in which the feature selection process is the most effective. On most cases where we have high dimensional data and small size of samples, the best performance results are achieved by ensemble methods. For the experimented data set with small dimensionality and high sample size, the best classification results are obtained by applying only a baseline feature selection algorithm. Ensemble methods are not efficient.

In the second part of the chapter, we proposed an ensemble feature selection approach based on feature selectors reliability assessment. We used simplicity and fastness of filters to create the selectors ensemble and obtain the best feature subsets among the whole feature space. Then, we proposed a robust aggregation technique based on multiple classifiers performance to combine selectors ensemble output. A classification algorithm was used as an evaluator to assign a confidence to features based on the feature subset associated classification performance to yield finally a reliability level of each selected feature. We compared our proposed approach to several existing techniques and to individual feature selection results and showed that our approach improves classification performance and stability for high dimensional and small sample size data sets or at least maintains the best individual results when they are specially high. To enhance stability, the data perturbation setting is better than the algorithm perturbation setting as it yields optimal stability results. The next chapter investigates prior knowledge some dimensions known to be more relevant, as a means of directing the feature selection process and boosting the feature selection stability.

Chapter 4

Prior Knowledge Based Feature Selection

Contents

4.1 Introduction

In many classification areas, experts may have prior knowledge about the relevance of some features. This constitutes a means to guide feature selection even if available knowledge concerns only a fraction of the features. Traditional feature selection algorithms tend to ignore prior knowledge about features.

In this chapter, we propose to study the effect of incorporating prior knowledge on feature selection stability and classification performance. First, we extend three well known feature selection methods, two filters and one embedded method, by incorporating prior knowledge about some dimensions known to be more relevant, as a means of directing the feature selection process. Our objective is to obtain a robust features subset that improves the selection stability and the classification performance.

Second, we propose a robust embedded feature selection method based on prior knowledge. This method makes use of a partial supervision on some features assumed a priori to be more relevant. Prior knowledge about these dimensions known to be more relevant is incorporated. Iteratively we make use of the initial prior knowledge and the previously selected features to expand a subset of highly relevant features in a pre-processing phase of feature selection.

4.2 Prior knowledge based extensions for stable feature selection

4.2.1 Incorporating prior knowledge in feature selection

In most feature selection applications, it is usually assumed that all features are equally relevant before the selection procedure. However, having prior knowledge about how features can be related to the prediction task will always help feature selection and its subsequent application. Hence, it is useful to use this information. For example, when the biological relevance of features can be proven, potentially relevant features can be favored and irrelevant ones can be eliminated. In many classification areas experts may have prior knowledge about some features which can bias the selection towards some features assumed to be more relevant. Prior knowledge is any information about features that can be used in feature selection to guide the selection process. He and Yu (2010) cited three sources of prior knowledge. It is either obtained from domain experts, relevant publications

or extracted from relevant data sets via transfer learning as investigated by Helleputte and Dupont (2009a). The incorporation of prior knowledge in the feature selection process can improve the classification performance and make the final SFS more stable. We are interested in approaches where prior knowledge can be used to explore the area in the feature space covered by pre-existing knowledge.

Zhao et al. (2008) integrated information from various data sources as prior knowledge to select genes from expression profiles. They use information contained in multiple data sources to extract an intrinsic global geometric pattern and use it in covariance analysis for gene selection. Taskar et al. (2003) use meta-features of words for text classification when there are features (words) that are unseen in the training set, but appear in the test set. In their work, features are words and meta-features are words in the neighborhood of each word. Other ideas using feature properties to produce or select good features can be found in the literature and have been applied in various applications. Lee et al. (2007) used transfer learning to construct an informative prior on feature relevance. They assumed that features themselves have meta-features that are predictive of their relevance to the prediction task from an ensemble of related prediction tasks sharing a similar relevance structure.

We propose to use prior knowledge obtained from domain experts and relevant publications about three high dimensional data sets to guide the selection process of two filters and an embedded feature selection algorithms. Proposed methods are discussed in the following sections.

4.2.2 Proposed prior knowledge based algorithms

In this section, we describe extensions of two filters and an embedded algorithm. These algorithms are mRMR, Relief and SVM.RFE. Our justification for choosing these algorithms is that they have been very popular and effective in the context of feature selection, and that we can integrate background knowledge on their feature selection process. The three proposed prior knowledge based methods are described below.

4.2.2.1 PK-mRMR

In this section, we present the prior knowledge based mRMR (PK-mRMR) as an extension of mRMR feature selection algorithm based on mutual information proposed by Peng et al. (2005).

Let M be a matrix containing a training data set $DS = (x_i, \omega_i)_{i=1}^{m}$, where x_i is the i^{th} data sample containing d features $(f_j)_{j=1}^{d}$, ω_i is its corresponding class label, and $d >> m$.The mRMR method selects a feature subset that has the highest relevance with the target class, subject to the constraint that selected features are mutually as dissimilar to each other as possible. Given f_j, representing the attribute j, and the class label ω, their mutual information is defined in terms of their frequencies of appearances $p(f_j)$, $p(\omega)$, and $p(f_j, \omega)$ as follows

$$I(f_j, \omega) = \int p(f_j, \omega) \log \frac{p(f_j, \omega)}{p(f_j)p(\omega)} df_j d\omega. \tag{4.1}$$

We incorporate the prior knowledge about each feature f_j by adding $\beta(f_j)$ to the mutual information as follows

$$I_{prior}(f_j, \omega) = I(f_j, \omega) + \beta(f_j). \tag{4.2}$$

The Maximum-Relevance method selects the best individual features correlated to the class labels by finding a feature set S with n features, which jointly has the largest dependency, $D(S, \omega)$, on the target class ω

$$\max D(S, \omega), D = \frac{1}{|S|} \sum_{f_j \in S} I_{prior}(f_j, \omega). \tag{4.3}$$

However, the correlations among those top features may be high. In order to remove the redundancy among features, a Minimum-Redundancy criterion, $minR(S)$, is introduced where mutual information between each pair of attributes is taken into consideration. This criterion is given by

$$\min R(S), R = \frac{1}{|S|^2} \sum_{f_j, f_u \in S} I(f_j, f_u). \tag{4.4}$$

By combining optimization criteria of Eqs. (4.3) and (4.4), mRMR improves the generalization properties of the features in the subset and the classification performance.
An incremental process is used to select features satisfying optimization criteria of Eqs. (4.3) and (4.4). Suppose that A represents the whole feature set and we have already selected the feature set S_{n-1}, with $n-1$ features. In order to choose the n^{th} feature from the set $\{A - S_{n-1}\}$, the two constraints D and R are combined and the feature maximizing this combination is selected as follows

$$\max_{f_j \in A - S_{n-1}} [I_{prior}(f_j, \omega) - \frac{1}{n-1} \sum_{f_j \in S_{n-1}} I(f_j, f_u)]. \tag{4.5}$$

4.2.2.2 PK-Relief

In this section, we present the prior knowledge based Relief algorithm (PK-Relief) as an extension of Relief feature selection algorithm developed by Kira and Rendell (1992). Relief algorithm assigns a relevance weight to each feature, which is meant to denote the relevance of the feature to the target concept. The algorithm samples instances randomly from the training set and updates the relevance values based on the difference between the selected instance and the two nearest instances of the same and opposite class. Kohavi and John (1997) demonstrated that since Relief randomly samples instances and their neighbors from the training set, its answers are unreliable without a large number of samples. In this study, we are concerned with high dimensional low sample size data. To avoid the randomized version of Relief constraint for this kind of data, we implemented a deterministic version of Relief that uses all instances. We integrate prior knowledge β_j about each feature f_j as described in Algorithm 3.

4.2.2.3 PK-RFE

In this section, we present the prior knowledge based RFE algorithm (PK-RFE) as an extension of RFE feature selection algorithm proposed by Guyon et al. (2002). RFE algorithm uses an iterative procedure to select features by training the classifier where the weights w_j are optimized with respect to a cost function I computed on training examples.

Algorithm 3 PK-Relief

Input:

$[\mathbf{X}, \beta, T]$

(T: a threshold to retain relevant features)

Find near-hit $NH(x_{ij})$, near-miss $NM(x_{ij})$

Calculate the margin of (x_{ij})

$w(x_{ij}) = d(x_{ij}, NM(x_{ij})) - d(x_{ij}, NH(x_{ij}))$

$W_j = \sum_{i=1}^{m} w(x_{ij})$

Integrate prior knowledge

$W_j = Beta_j + W_j$

if $W_j >= T$ **then**

add f_j to selected-features

end if

return selected features

Then, the ranking criterion is computed for all features based on w_j^2. This process is iterated and the feature with the smallest ranking criterion is removed. The remaining features are selected. This iterative procedure is a backward feature elimination (Kohavi and John (1997)). SVM-RFE proposed by Guyon et al. (2002) is an application of RFE using weight magnitude as the ranking criterion.

We propose PK-RFE which consists of incorporating the vector of prior knowledge β with feature weights obtained after applying an SVM classifier on the training set. Let $\beta_j \geq 1$ denote the relative prior relevance of the j^{th} feature. The relevance value for each component β_j is arbitrarily assigned a value of 10 if the feature $\mathbf{f}_j$ is a priori relevant and a value of 1 otherwise. The PK-RFE algorithm has the following four steps:

- Train an SVM on the training set and obtain a vector of feature weights;
- Consider the vector made of absolute values of each SVM feature and multiply it component-wise by the corresponding dimension of β the vector of prior relevance.
- Normalize this vector to a unit-norm and multiply the input data component-wise by this vector.

- Iterate until convergence.

Along the various iterations, some dimensions drop to zero and the remaining ones are the selected features influenced by the prior relevance.

4.2.3 Experimental study

In this section we report the experimental setup and results of our proposed feature selection methods. These methods are applied to several microarray data sets described below. Three evaluation metrics, namely the classification performance, the stability of the selected genes and the McNemar's statistical test are defined respectively.

4.2.3.1 Datasets and prior knowledge

Three high dimensional data sets are used in our experimental study, namely DLBCL, Bladder and Lung cancer data sets. These data sets are defined in Chapter 2.

For the Bladder cancer dataset, a list of eleven a priori relevant features, markers, are collected from the literature, looking systematically at the Pubmed literature on markers of recurrence and progression of bladder cancer.

Shipp et al. (2002) mention two genes as clinical markers to discriminate DLBCL tissues from Follicular Lymphomas: Transferrin Receptor and Lactate Dehydrogenase A.

For Lung cancer data set, Guan et al. (2009) collected prior knowledge from any proven information about lung adenocarcinoma related genes in the literature. They restricted their attention to the journal entitled "Cancer Research". Cancer Research's publication scope covers all subfields of cancer research. The full texts of the papers were downloaded and then lung adenocarcinoma-related genes were retrieved from the literature. Then, after these genes' locations in the original dataset were collected, the genes were tested through multiple testing procedure in the training set provided by Gordon et al. (2002). Eight significant genes were retained.

Table (4.1) summarizes the characteristics of the three data sets, namely the number of samples, the initial dimension of the input space and the number of a priori relevant features.

Table 4.1: Datasets characteristics

Dataset	# samples	# features	# a priori relevant features
Bladder cancer	31	3036	11
DLBCL	77	7029	2
Lung cancer	181	12533	8

4.2.3.2 Performance metrics

Classification performance: We use 10-fold stratified CV to predict the classification performance of SVM classifier with our three proposed techniques and their original versions on three data sets with the selected feature sets. We use the stability index proposed by Kuncheva (2007) to measure stability of our feature selection methods.

Statistical evaluation : The McNemar's test

The experimental study compares proposed algorithms to their original versions. For this reason, classification accuracy is important and to be more precise, we use this the McNemar's statistical test (Eveitt (1977)). This metric is applied to test whether the proposed algorithm significantly outperforms others on these data sets. For a data set, the McNemar's statistical test compares algorithms A and B based on the following values:

- $N00$: number of test data misclassified by both algorithms A and B
- $N01$: number of test data misclassified by algorithm A but not B
- $N10$: number of test data misclassified by algorithm B but not A
- $N11$: number of test data misclassified by neither algorithms A nor B

Under the null hypothesis, the two algorithms should have the same error rate, which means that $N01 = N10$. McNemar's test is based on a χ^2 test for goodness of fit that compares the distribution of counts expected under the null hypothesis and the observed counts. The statistic given by:

$$\frac{(|N01 - N10| - 1)^2}{N01 + N10} \tag{4.6}$$

is distributed approximately as χ^2 with 1 degree of freedom. The null hypothesis is accepted where this quantity is less than $\chi^2_{1,0.95} = 3.841459$ or with a p-value greater than 0.05 (Dietterich (1998)). Otherwise, we reject the null hypothesis in favor of the alternative hypothesis that the two algorithms have different performance when trained on the particular training set.

Based on these three evaluation criteria, we compare our proposed algorithms with their original versions, mRMR, Relief and SVM.RFE, which does not integrate prior knowledge into the feature selection process.

4.2.3.3 Results analysis

Classification performance and statistical results:

Tables (4.2) to (4.4) present the results of applying the three proposed algorithms on the three data sets, where N denotes the number of selected features.

We highlighted the best performance of each algorithm in order to make comparison easier. Table (4.2) shows that for Bladder cancer data set, the best classification accuracy is $96,8\%$. It is obtained by PK-RFE algorithm with a minimum selected subset cardinality of 40 features. This performance remains stable for higher feature subset sizes and decreases for a cardinality of 90. RFE algorithm gives $93,5\%$ with 10 features. The best classification accuracy of mRMR is $93,5\%$ with 90 features. PK-Relief follows with $90,3\%$ obtained with 30 features. For DLBCL data set, Table (4.3) shows that PK-RFE is also performing with $97,4\%$ accuracy achieved with 70 features. It is followed by RFE. In the third place, we have PK-mRMR and mRMR. PK-Relief and Relief achieve also the same performance for this data set. Table (4.4) shows that 100% accuracy is obtained by a subset of 40 features selected by PK-Relief algorithm for Lung cancer data set. PK-RFE and RFE give the same perfect classification performance with higher feature subset sizes. According to these classification results, the best classification performances for the three data sets are obtained by a PK based approach. PK-RFE is the best in two out of three cases. It is expected that PK-RFE and RFE perform better than other algorithms as they are embedded methods. They use the bias of the SVM algorithm to select features and thus perform better. From a general point of view, classification results are often similar for the original feature selection methods and their proposed extensions.

Table 4.2: Classification performance and McNemar's statistical test on Bladder cancer data set.

N	Relief	PK-Relief	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.871	0.839	0.617 - 0.25
20	0.806	0.871	0.479 - 0.5
30	0.806	**0.903**	0.371 - 0.8
40	0.871	0.903	1 - 0
50	0.839	0.903	0.617 - 0.25
60	0.806	0.903	0.371 - 0.8
70	0.806	0.871	0.617 - 0.25
80	0.806	0.871	0.617 - 0.25
90	**0.903**	0.839	0.479 - 0.5
100	0.871	0.903	1 - 0
N	mRmr	PK-mRmR	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.774	0.871	0.617 - 0.25
20	0.839	0.806	0.479 - 0.5
30	0.806	0.806	0.479 - 0.5
40	0.839	0.806	1 - 0
50	0.839	0.806	1 - 0
60	0.839	0.839	0.479 - 0.5
70	0.839	0.839	0.479 - 0.5
80	0.903	**0.903**	0.479 - 0.5
90	**0.935**	0.903	1 - 0
100	0.903	0.903	0.479 - 0.5
N	RFE	PK-RFE	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.935	0.903	1 - 0
20	0.903	0.935	1 - 0
30	0.903	0.935	1 - 0
40	**0.935**	**0.968***	1 - 0
50	0.935	0.968	1 - 0
60	0.903	0.968	0.479 - 0.5
70	0.903	0.968	0.479 - 0.5
80	0.935	0.968	1 - 0
90	0.935	0.935	0.479 - 0.5
100	0.903	0.935	1 - 0

Table 4.3: Classification performance and McNemar's statistical test on DLBCL cancer data set.

N	Relief	PK-Relief	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.831	0.792	0.45 - 0.571
20	0.896	0.896	0.479 - 0.5
30	0.870	0.857	1 - 0
40	0.896	0.896	0.479 - 0.5
50	0.909	0.909	0.479 - 0.5
60	0.909	0.909	0.479 - 0.5
70	0.909	0.909	0.479 - 0.5
80	**0.922**	**0.922**	0.479 - 0.5
90	0.922	0.922	0.479 - 0.5
100	0.922	0.922	0.479 - 0.5
N	mRMR	PK-mRMR	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.909	0.922	0.683 - 0.167
20	0.935	0.896	0.617 - 0.25
30	0.87	0.883	1 - 0
40	0.935	0.922	1 - 0
50	0.922	0.909	1 - 0
60	0.935	0.935	0.479 - 0.5
70	0.935	0.935	0.479 - 0.5
80	**0.948**	**0.948**	0.479 - 0.5
90	0.935	0.948	1 - 0
100	0.909	0.922	1 - 0
N	RFE	PK-RFE	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.935	0.844	0.045 - 4
20	0.948	0.9091	0.371 - 0.8
30	**0.961**	0.922	0.371 - 0.8
40	0.961	0.935	0.479 - 0.5
50	0.948	0.909	0.248 - 1.333
60	0.948	0.9610	1 - 0
70	0.948	**0.974***	0.479 - 0.5
80	0.948	0.948	0.479 -0.5
90	0.948	0.974	0.479 - 0.5
100	0.948	0.961	1 - 0

Table 4.4: Classification performance and McNemar's statistical test on Lung cancer data set.

N	Relief	PK-Relief	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.945	0.978	0.181 - 1.786
20	0.961	0.968	1 - 0
30	0.961	0.972	0.683 - 0.167
40	0.983	**1***	0.248 - 1.333
50	0.972	0.989	0.248 - 1.333
60	0.978	0.983	1 - 0
70	0.978	0.994	0.248 - 1.333
80	0.983	0.989	1 - 0
90	0.983	0.989	0.479 - 0.5
100	**0.994**	0.994	0.479 - 0.5
N	mRMR	PK-mRMR	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.901	0.983	0.001 - 11.529
20	0.989	0.983	1 - 0
30	0.983	0.978	1 - 0
40	0.983	0.983	0.6171 - 0.25
50	0.989	0.983	1 - 0
60	**0.995**	0.989	1 - 0
70	0.995	**0.995**	0.4795 - 0.5
80	0.989	0.995	1 - 0
90	0.989	0.989	0.479 - 0.5
100	0.989	0.994	1 - 0
N	RFE	PK-RFE	McNemar's test
	PCA	PCA	p-value - χ^2
10	0.978	0.983	1 - 0
20	0.994	0.978	0.248 - 1.333
30	0.989	0.995	1 - 0
40	0.995	0.989	1 - 0
50	**1**	0.989	0.479 - 0.5
60	1	**1**	0.479 - 0.5
70	1	1	0.479 - 0.5
80	1	1	0.479 - 0.5
90	1	0.989	0.479 - 0.5
100	1	0.989	0.479 - 0.5

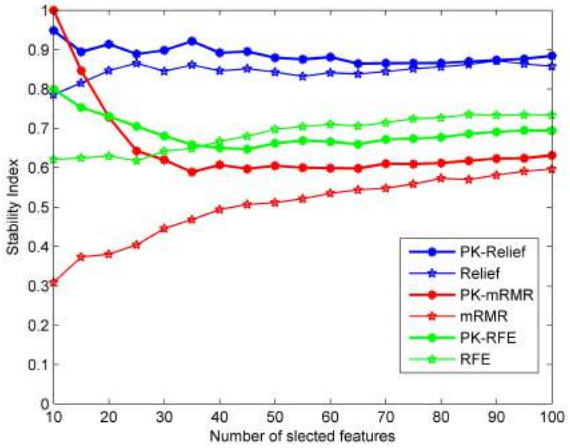
The statistical results using McNemar's test of the PK based algorithms show that their performances are not significantly different from the original ones as McNemar's test is often less than 3.841459 and p-values are higher that the 5% significance level. It is interesting to compare the stability performance.

Stability results:

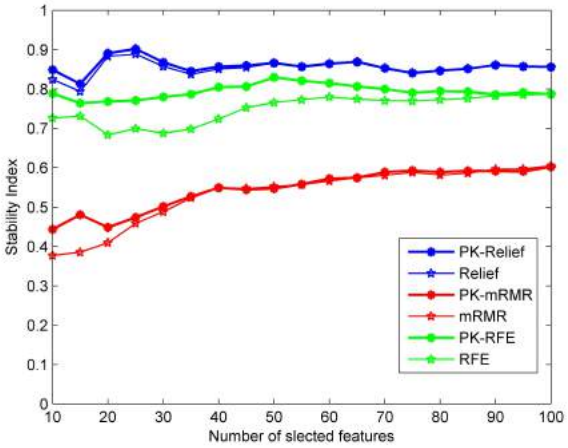
Figure (4.1.a) shows stability results with Kuncheva index calculated on 10 folds of Bladder cancer data set. The stability rate is perfect and equal to 1 with selected subsets of cardinality 10 for each fold by PK-mRMR algorithm. For the same considered cardinality mRMR performs a poor stability of about 0,3. mRMR becomes more stable when the number of selected features increases, in contrast with PK-mRMR. This may be explained by the effect of prior knowledge on selecting the first features. Nevertheless, stability of PK-mRMR remais higher for higher feature subsets cardinalities. For subsets of 10 features in Lung cancer data set, PK-RFE stability is about 0,95, however RFE stability is about 0,68, as shown in Figure (4.1.c). Stability results of Relief and PK-Relief are often similar for all data sets. We can say that stability of PK based algorithms is better than classical algorithms versions specially for the first experimented cardinalities. Stability measures become similar by increasing feature subset cardinalities. We deduce that the incorporation of background knowledge guides feature selection and may affect stability. This is important for feature selection applications as it increases the confidence of discovered features.

4.2.4 Discussion

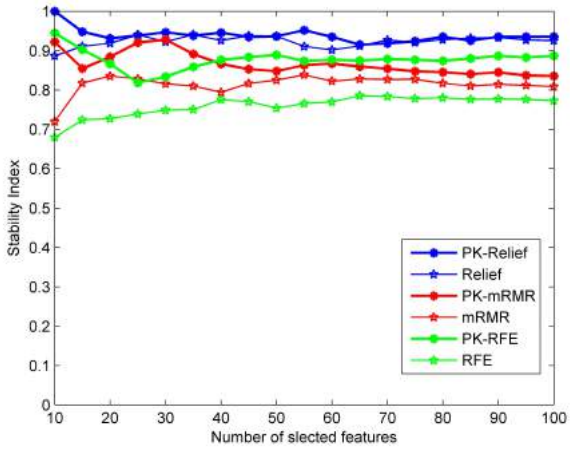
We investigated research on the effect of integrating background knowledge about some dimensions known to be more relevant, as a means of directing the feature selection process. We proposed extensions of two filters and an embedded feature selection technique, by incorporating prior knowledge in the search procedure of the most relevant features. Our objective was to guide feature selection by prior knowledge in order to obtain a stable result as a good set of features is ideally highly stable with respect to sampling variation. We studied the effect of the proposed techniques on the classification performance and the stability of the feature selection and compared them with their original versions, which do not integrate prior knowledge. Results show that integrating prior knowledge increased stability in some cases compared to classical approaches. Also, our proposed techniques



(a) Bladder Cancer data set



(b) DLBCL data set



(c) Lung cancer data set

Figure 4.1: Feature selection stability with Kuncheva Index.

often outperform other methods in terms of classification accuracy. However, statistical results prove that the performance improvements are modest. It is interesting to search for a new approach which uses a more sophisticated way of integrating prior knowledge in the feature selection process to obtain better classification and stability performances. We propose such method in the following.

4.3 Stable feature selection based on semi supervised relevance learning

One of the existing prior knowledge based feature selection methods, is the partially-supervized-$l2$-AROM algorithm (PS-$l2$-AROM) proposed by Helleputte and Dupont (2009b). In their work, the algorithm integrates prior knowledge about some genes known as clinical markers to discriminate DLBCL tissues from Follicular Lymphomas. Before the feature selection process, they assign a relevance value for those genes assumed to be more relevant. PS-AROM methods modify a linear model objective function, called $l1$-AROM described by Weston et al. (2003), by adding a prior feature relevance vector $\beta = [\beta_1, ..., \beta_d]$ defined over the input dimensions. The optimization problem of PS-$l2$-AROM penalizes the least those dimensions which are assumed a priori more relevant and thus guides the feature selection process. Iteratively, an objective function is solved given the previous features weight vector w along with the fixed relevance vector β, and the process is iterated till convergence. The original $l2$-AROM method is obtained when $\beta_j = 1$, $\forall$ feature f_j, in other words, without prior preference between input features. While in the method proposed by Helleputte and Dupont (2009b) the feature selection algorithm is modified by integrating prior knowledge only once in the feature selection process, in our proposed approach prior knowledge is expanded and integrated iteratively into the feature selection algorithm. Our formulation adopts a more advanced framework which takes advantage of prior knowledge to search in a first step for more relevant features based only on their neighborhood with features assumed a priori relevant. Then, in a second step the extended set of a priori relevant features is integrated in the feature selection which gives the final feature subset.

4.3.1 Proposed approach: Semi-Supervised-$l2$AROM

We exploit prior knowledge on feature relevance in an iterative two step approach. In the first step we extend the set of relevant features using a semi-supervised approach. Our basic assumption here is that the feature set of a-priori known to be relevant features is not complete, thus we include in it features that are similar to the relevant features. In the second step we use a feature selection algorithm that exploits knowledge on feature relevance to supervise feature selection. The two steps are iterated until convergence, i.e. until there is no change in the set of relevant features. We call the proposed approach the

Semi-Supervised-$l2$AROM (SS-$l2$AROM).

Let X be a matrix containing m instances $\mathbf{x}_i = (x_{i1}, \ldots, x_{id}) \in \mathbb{R}^d$, where d is the number of features, and $\mathbf{y}_i = (y_1, \ldots, y_m), i = 1, \ldots, m$ the vector of class labels for the m instances. Let A be the set of features $\mathbf{f}_j = (f_1, \ldots, f_d), j = 1, \ldots, d$ where $d >> m$. We denote by $R_n \subseteq A$ the set of features that are known to be relevant based on prior knowledge at iteration n. $\beta = [\beta_1, \ldots, \beta_d]$ is a vector of background knowledge about the input dimensions. The higher the value of β_j the more relevant the corresponding feature is a priori assumed.

Our proposed approach consists initially of solving a semi supervised problem where the training set is given by $\mathbf{X}'$ the transpose of $\mathbf{X}$, i.e. the j^th row is $f_j = (f_{j1}, \ldots, f_{jm})$. The feature f_j is labeled as Relevant (1) if $f_j \in R_n$ and Unknown (0) otherwise. After this step, and to extend the set of relevant features, an additional set of features predicted as relevant P_n is obtained and added to R_n such that $R'_n = R_n \cup P_n$. The second step of our approach consists of applying on the original matrix $\mathbf{X}$ a feature selection algorithm that can handle prior knowledge on feature relevance using R'_n as the set of a priori relevant features. This step yields a new selected feature set denoted by R_{n+1}. This process is iterated until the number of final selected features reaches a desired feature set cardinality. The proposed algorithm is detailed below.

4.3.1.1 First phase: Semi-supervised relevance learning

In pattern recognition applications, the training data is assumed to be appropriate with the underlined problem. For the purpose of our semi supervised problem, aiming at predicting new relevant features using a priori relevant ones, we proceed with data transformation in order to make it fit the problem. Initially, we take the transpose of the data matrix $\mathbf{X}$ in such way that features become the training instances. Then, each feature is assigned a label indicating whether it is a priori relevant (Relevant (1)) or (Unknown (0)).

In the first stage of the nth iteration we solve a semi-supervised problem where we are given a vector, β_n, which describes whether a feature is known to be relevant or not, to find additional relevant features if they exist. This is conducted by applying some semi-supervised algorithm which returns an updated feature relevance vector β'_n.

Now, using a kNN algorithm, distances are calculated between a priori relevant features

Algorithm 4 Semi-supervised relevance learning

Input:

$[\mathbf{X}^T, \beta_n, R_n]$

$R'_n = R_n$

$\forall f_j \in \overline{R_n}$

$\mathbf{S}_{f_j}$ = kNN(f_j)

if $\forall g \in \mathbf{S}_{f_j}, g$ is relevant **then**

$R'_n := R'_n \cup f_j$, i.e. f_j is relevant

β'_n=Update($[\beta_n, R'_n]$)

end if

return R'_n, β'_n

and the remaining features. For each feature f_j, which is part of the features for which we do not know whether they are relevant or not, i.e. $f_j \in \overline{R_n}$, we need to find the set of its k nearest neighbors which we denote by $\mathbf{S}_{f_j}$. If all its nearest neighbors are known to be relevant then we denote also f_j as relevant. It is very important that the semi-supervised algorithm is well-behaved, i.e. it will not continue producing relevant features in a trivial way until we get the full feature set. However, it will stop at some point. Features found to be relevant by the semi-supervised algorithm are used to extend R_n to R'_n. The vector of prior knowledge β_n is updated to β'_n based on the a priori relevant feature subset R'_n such that each component of this vector is assigned a value of 10 if a feature $\mathbf{f}_j \in R'_n$, and a value of 1 otherwise. The algorithm for the first phase is given in Algorithm 4.

4.3.1.2 Second phase: Application of feature selection algorithm

In the second step of our method we deploy a feature selection algorithm on the original data matrix X and the class labels, which is able to incorporate domain knowledge on features that are known to be relevant. We will use PS-$l2$-AROM (Helleputte and Dupont (2009b)) as the algorithm of this second step. This method is based on an embedded selection method with linear models, called $l2$-AROM (Weston et al. (2003)). The steps of the algorithms are:

- At step $k = 0$, initialize $w_k = \beta$

- iterate until convergence:
- $min_w \parallel w \parallel_2^2$ subject to: $y_i(w(x_i * w_k) + b) \geq 1$
- let $(\overline{w})$ be the solution, set $w_{k+1} \leftarrow w_k * \overline{w} * \beta$

AROM methods are described in Section (1.2.3). PS-$l2$-AROM algorithm is applied in the second step of our algorithm. Iteratively the minimization problem in PS-$l2$-AROM algorithm is solved given the relevance vector β_n obtained in the first step. Iterations terminate when there are no important differences between the features indicated as relevant in step n by the vector β_n and the ones indicated as relevant in the step $n+1$ by β_{n+1}.

A crucial point here is whether there is a monotonic increase in R_n vector, i.e. as we move from step n to $n+1$, do we always have $R_n \subseteq R_{n+1}$? This obviously depends on the behavior of the semi-supervised learning and the feature selection algorithm that we have selected for stages one and two. So we need to study the convergence behavior of the two-step algorithm. This means that we should trace the convergence as a function of n as follows:

$$Conv = \frac{|R_n \cap R_{n+1}|}{|R_n \cup R_{n+1}|} \tag{4.7}$$

This quantity is equal to zero when there are no common features between iteration $n+1$ and n and to 1 when there is no difference between the feature sets selected respectively in iteration $n+1$ and iteration n, meaning that the algorithm has converged. Basically, this means that at some point the semi-supervised algorithm does not produce anymore additional relevant features, or produces very few ones. The algorithm's convergence is used as a stopping criterion for the feature selection process.

At the semi-supervised step we retrieve $|R'_n|$ features. Then, at the feature selection step we allow the feature selection algorithm to select at least as many features as possible such that $|R_{n+1}| \geq |R'_n|$. In order to control the number of features between the semi-supervised step and the feature selection step, we set the number of features to select at each step as follows: $|R_{n+1}| = (1+p) \times |R'_n|$ i.e. the number of features that are retained in the feature selection step should be as many as the ones in R'_n plus one small percentage, p. The algorithm is described in Algorithm 5.

Algorithm 5 Feature Selection with Background Knowledge

Input:

$\mathbf{X}$: an $m \times d$ dataset

$\mathbf{y}$: m-length vector of class labels

R_0: set of a priori relevant features.

β_0: d-length vector characterizing features as a-priori (10) relevant or not-known (0)

p: percentage of additional features to include in each step of the iteration at the feature selection step.

ϵ: tolerance variable determining when the algorithm converges: should be set to a small value, e.g. 0.01.

$n = 0$

$R_n = R_0$

$\beta_n = \beta_0$

repeat

$[R'_n, \beta'_n] = \text{SemiSup}([\mathbf{X}^T, \beta_n, R_n])$

$k = (1 + p) \times |R'_n|$ (number of features to select)

$R_{n+1} = PS - l2 - AROM([\mathbf{X}, \mathbf{y}], \beta'_n, k)$

$n = n + 1$

until $Conv \leq \epsilon$

4.3.2 Experimental study

In this section we report the experimental setup and results of our proposed feature selection method. This method is applied to several microarray data sets described in Section 4.2.3.1. Four evaluation metrics, namely the algortithm convergence test, the classification performance, the McNemar's statistical test and the stability of the selected features using Kuncheva Index are evaluated.

We first evaluate the convergence of SS-$l2$AROM. Then, based on the other evaluation criteria, we compare our proposed algorithm with two feature selection algorithms: PS-$l2$-AROM (Helleputte and Dupont (2009b)), described before, which considers prior knowledge and SVM.RFE (Guyon et al. (2002)), which does not integrate prior knowledge into the feature selection process.

4.3.2.1 Feature set evolution

A crucial point to consider in evaluating our proposed feature selection algorithm is to have a monotonic increase in the selected features (the R_n vector), i.e. as we move from step n to $n+1$ do we always have $R_n \subseteq R_{n+1}$? This obviously depends on the behavior of the semi-supervised learning and the feature selection algorithm that we have selected for stages one and two. Tables (4.5) - (4.7) give the results concerning the study of the convergence behavior of the two-step algorithm measured by the quantity $Conv$ defined above. For each fold of the CV, we measure the algorithm's convergence score for each iteration in the three data sets with a cardinality of features equal to 100.

The convergence scores show that for the three data sets, SS-$l2$AROM feature selection algorithm converges since the selected feature set becomes stable after a maximum of six iterations for Bladder cancer data set, ten iterations for DLBCL data set and a maximum of eight iterations for Lung cancer data set.

Table 4.5: Feature set convergence on Bladder cancer with SS-$l2$AROM.

Bladder : FS convergence						
Iteration	1	2	3	4	5	6
Fold1	0	0.6000	0.4599	0.9231	1	-
Fold2	0	0.5625	0.5038	0.9608	1	-
Fold3	0	0.3889	0.8868	0.9802	0.9802	1
Fold4	0	0.6260	0.5385	0.8182	0.9802	1
Fold5	0	0.5504	0.9231	0.6260	0.9048	1
Fold6	0	0.5267	0.4493	0.8519	1	-
Fold7	0	0.6807	0.8519	0.6949	0.9608	1

4.3.2.2 Results analysis

The proposed algorithm, SS-$l2$AROM, is compared with PS-$l2$-AROM and SVM.RFE. Table (4.8) presents the results of applying the three algorithms on the three data sets, where N denotes the number of selected features.

Classification and stability results

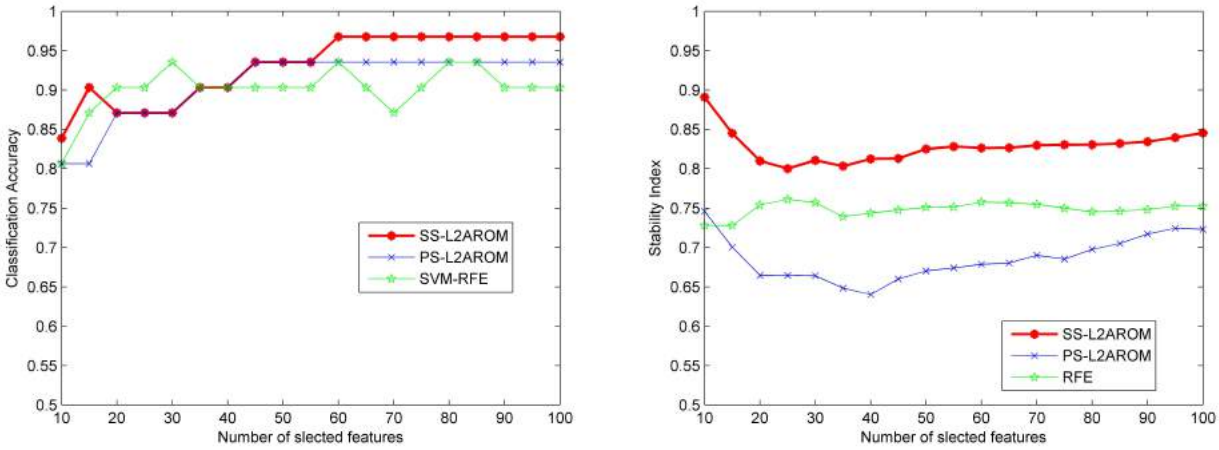
Table 4.6: Feature set convergence on DLBCL with SS-$l2$AROM.

DLBCL : FS convergence										
Iteration	1	2	3	4	5	6	7	8	9	10
Fold1	0	0.626	0.77	0.942	0.923	0.961	1	-	-	-
Fold2	0	0.613	0.852	0.905	0.942	1	-	-	-	-
Fold3	0	0.587	0.802	0.905	0.942	0.961	0.98	0.98	0.98	1
Fold4	0	0.639	0.786	0.887	0.942	1	-	-	-	-
Fold5	0	0.613	0.835	0.905	0.98	0.98	0.98	1	-	-
Fold6	0	0.667	0.818	0.923	1	-	-	-	-	-
Fold7	0	0.639	0.835	0.905	0.923	0.961	0.98	1	-	-
Fold8	0	0.667	0.786	0.905	0.961	0.98	1	-	-	-
Fold9	0	0.739	0.818	0.887	0.942	0.961	0.98	1	-	-
Fold10	0	0.681	0.852	0.905	0.942	0.98	0.98	1	-	-

Table 4.7: Feature set convergence on Lung cancer with SS-$l2$AROM.

Lung : FS convergence								
Iteration	1	2	3	4	5	6	7	8
Fold1	0	0.7391	0.8692	0.9608	0.9231	0.9802	0.9802	1
Fold2	0	0.7699	0.8519	1	-	-	-	-
Fold3	0	0.7391	0.8692	0.9608	0.9802	0.9802	0.9802	1
Fold4	0	0.7544	0.8519	0.9608	0.9802	1	-	-
Fold5	0	0.7857	0.8519	0.9231	0.9608	0.9802	0.9802	1
Fold6	0	0.7094	0.8182	0.9417	0.9608	0.9608	1	-
Fold7	0	0.7391	0.8182	0.9231	0.9231	1	-	-
Fold8	0	0.7544	0.8018	0.9231	0.9417	0.9608	1	-
Fold9	0	0.7241	0.8018	0.8692	0.9802	1	-	-
Fold10	0	0.6667	0.8182	0.9802	0.9608	0.9802	1	-

For Bladder cancer data set, SS-$l2$AROM gives the best classification performance with the best result obtained with a subset of 60 selected features (96.77%). SS-$l2$AROM gives also excellent stability results compared to PS-$l2$AROM and SVM-RFE. The best stability value is 0.8908 obtained with 10 features.



(a) Classification performance

(b) Stability

Figure 4.2: Classification performance and feature selection stability with Kuncheva Index on Bladder cancer data set.

For DLBCL and Lung cancer data sets, SS-$l2$AROM still yields the best classification results as noticed in Figure (4.3.a) and Figure (4.4.a). PS-$l2$AROM is a competitor algorithm concerning classification results both on DLBCL and Lung cancer data sets, its stability is smaller but similar to SS-$l2$AROM on DLBCL. However, the outperformance of the latter algorithm is clearly visible on Lung cancer data set. The stability behaviour of SVM-RFE is not the same for the three data sets and it is specially modest for DLBCL data set. Thus, in most cases prior knowledge improves classification performance and stability results.

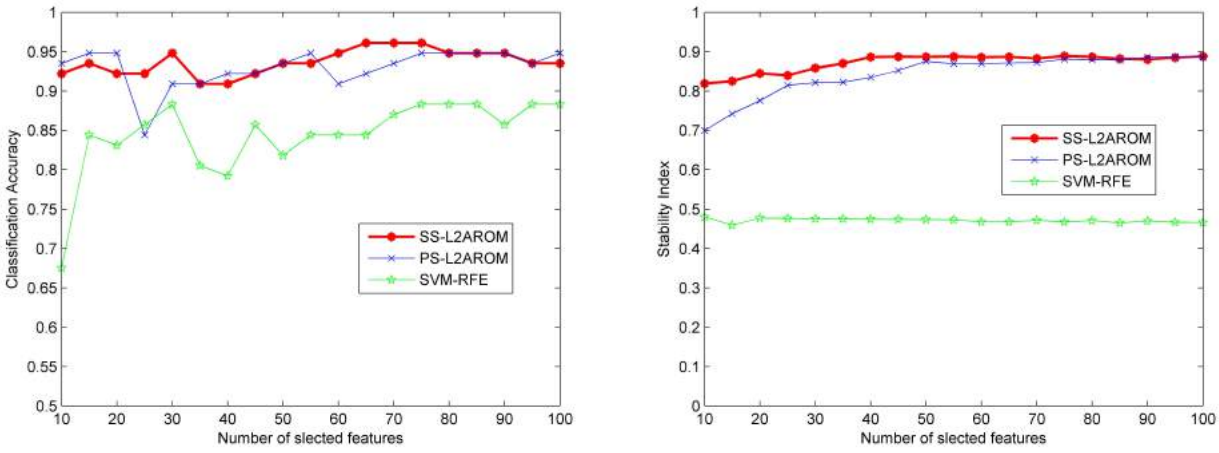
McNemar's test results

In Table (4.9), the arrowheads ($\leftarrow$) denote which algorithm performed better for the given data sets with a feature set cardinality of 100 features. McNemar's test values are given next to the arrowheads as a measure of how significant the results are.

The McNemar's test results of SS-$l2$AROM are always advantageous, but according

Table 4.8: Classification performance coupled with feature selection stability on Bladder cancer, DLBCL and Lung cancer data sets.

Bladder cancer			
N	SS-$l2$AROM	PS-$l2$AROM	SVM-RFE
10	83.87- 0.8908	80.65- 0.7458	80.65- 0.7280
20	87.10- 0.8099	87.10- 0.6645	90.32- 0.7539
30	87.10- 0.8107	87.10- 0.6641	93.55- 0.7569
40	90.32- 0.8125	90.32- 0.6403	90.32- 0.7438
50	93.55- 0.8251	93.55- 0.6701	90.32- 0.7510
60	96.77- 0.8264	93.55- 0.6786	93.55- 0.7575
70	96.77- 0.8297	93.55- 0.6900	87.10- 0.7547
80	96.77- 0.8305	93.55- 0.6976	93.55- 0.7452
90	96.77- 0.8343	93.55- 0.7170	90.32- 0.7481
100	96.77- 0.8454	93.55- 0.7229	90.32- 0.7523
DLBCL			
N	SS-$l2$AROM	PS-$l2$AROM	SVM-RFE
10	92.21 - 0.8188	93.51 - 0.6996	67.53 - 0.4802
20	92.21 - 0.8448	94.81 - 0.7756	83.12 - 0.4771
30	94.81 - 0.8581	90.91 - 0.8215	88.31 - 0.4756
40	90.91 - 0.8863	92.21 - 0.8348	79.22 - 0.4744
50	93.51 - 0.8868	93.51 - 0.8753	81.82 - 0.4734
60	94.81 - 0.8855	90.91 - 0.8695	84.42 - 0.4677
70	96.10 - 0.8825	93.51 - 0.8722	87.01 - 0.4717
80	94.81 - 0.8868	94.81 - 0.8796	88.31 - 0.4708
90	94.81 - 0.8800	94.81 - 0.8853	85.71 - 0.4700
100	93.51 - 0.8880	94.81 - 0.8851	88.31 - 0.4663
Lung cancer			
N	SS-$l2$AROM	PS-$l2$AROM	SVM-RFE
10	98.34 - 0.8065	98.90 - 0.6508	91.16 - 0.7109
20	99.45 - 0.7941	99.45 - 0.7329	93.92 - 0.7552
30	99.45 - 0.8396	99.45 - 0.8099	96.13 - 0.7282
40	100 - 0.8250	99.45 - 0.7826	95.03 - 0.7191
50	100 - 0.8385	100 - 0.7680	93.37 - 0.7242
60	100 - 0.8605	100 - 0.7825	94.48 - 0.7302
70	100 - 0.8602	100 - 0.7874	95.58 - 0.7353
80	100 - 0.8541	100 - 0.7853	93.92 - 0.7408
90	100 - 0.8565	100 - 0.7789	96.13 - 0.7391
100	100 - 0.8640	100 - 0.7912	97.24 - 0.7377



(a) Classification performance

(b) Stability

Figure 4.3: Classification performance and feature selection stability with Kuncheva Index on DLBCL cancer data set.

to Dietterich (1998), performances of SS-$l2$AROM and PS-$l2$AROM are not significantly different. However SS-$l2$AROM outperforms significantly SVM-RFE on two out of three data sets as McNemar's test is larger than 3.841459 and p-values are respectively 0.02 and 0.003 which means that the null hypothesis is rejected at 5% significance level.

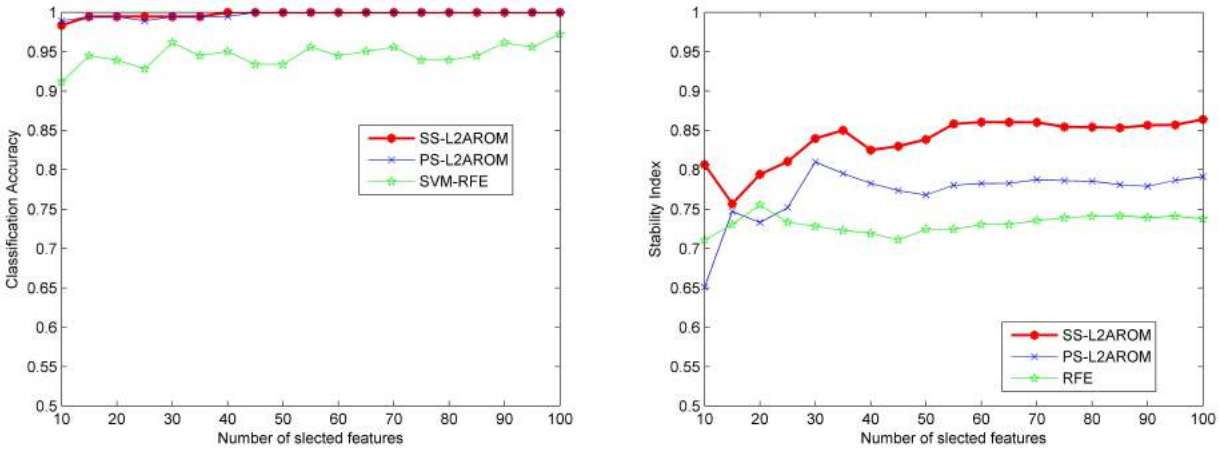
From this empirical study, we deduce that algorithms which incorporate prior knowledge have a better classification accuracy than the other feature selection algorithms. This is not always the case for the stability of feature selection, but our proposed method, namely SS-$l2$AROM, is also advantageous in this respect. Consequently, considering background knowledge about features is very important and beneficial to guide the feature selection process. Moreover, taking advantage of this prior knowledge to extend the set of a priori relevant features in a pre-processing phase of feature selection further improves both classification and feature selection stability.

4.3.3 Discussion

We propose a robust feature selection method, SS-$l2$AROM, based on semi supervised prior relevance learning. Prior knowledge about some dimensions known to be more relevant is incorporated as a means of guiding the feature selection process. The objective is to make use of a partial supervision on features assumed a priori to be more relevant, in

Table 4.9: McNemar's test results.

Bladder cancer data set			
	SS-$l2$AROM	PS-$l2$AROM	SVM-RFE
	p-value - χ^2	p-value - χ^2	p-value - χ^2
SS-$l2$AROM	-	$\leftarrow [0.5 - 0.25]$	$\leftarrow [0.5 - 0.25]$
PS-$l2$AROM	-	-	$\leftarrow [0.24 - 0.125]$
SVM-RFE	-	-	-
DLBCL data set			
	SS-$l2$AROM	PS-$l2$AROM	SVM-RFE
	p-value - χ^2	p-value - χ^2	p-value - χ^2
SS-$l2$AROM	-	$\leftarrow [0.124 - 2.08]$	$\leftarrow [\mathbf{0.02 - 5.04}]$
PS-$l2$AROM	-	-	$\leftarrow [0.185 - 1.25]$
SVM-RFE	-	-	-
Lung cancer data set			
	SS-$l2$AROM	PS-$l2$AROM	SVM-RFE
	p-value - χ^2	p-value - χ^2	p-value - χ^2
SS-$l2$AROM	-	$\leftarrow [0.5 - 0.25]$	$\leftarrow [\mathbf{0.003 - 8.03}]$
PS-$l2$AROM	-	-	$\leftarrow [\mathbf{0.013 - 5.625}]$
SVM-RFE	-	-	-



(a) Classification performance

(b) Stability

Figure 4.4: Classification performance and feature selection stability with Kuncheva Index on Lung cancer data set.

order to select a robust feature set in an interactive manner. Iteratively we make use of the initial prior knowledge and the previously selected features to learn new relevant features by a semi supervised approach. The extended subset of relevant features is used as prior knowledge to be integrated in a second step to guide the feature selection process until an optimal number of features is obtained. Our proposed approach shows encouraging results both for improving the classification accuracy and for dealing with the instability problem in feature selection for high dimensional data. Experiments on three microarray data sets show that the partial supervision in SS-$l2$AROM improves both classification and stability performances compared to PS-$l2$AROM and SVM-RFE. Our proposed approach fits with any feature selection algorithm that can integrate prior knowledge.

4.4 Conclusion

In this chapter, we investigated the effect of integrating background knowledge about some dimensions known to be more relevant, as a means of directing the feature selection process. We proposed extensions of two filters and an embedded feature selection technique, by simply incorporating prior knowledge in the search procedure of the most relevant features. We studied the effect of the proposed techniques on the classification performance

and the stability of the feature selection and compared them with their original versions, which do not integrate prior knowledge. The results showed that integrating prior knowledge increased stability in some cases compared to classical approaches. However, the effect of classification accuracy is not noticed.

We proposed a more advanced feature selection method, SS-$l2$AROM, based on semi supervised prior relevance learning. The objective of this approach is to make use of a partial supervision on features assumed a priori to be more relevant, in order to select a robust feature set in an interactive manner. This approach improved classification accuracy compared to SVM.RFE and increased stability of feature selection for high dimensional data compared to two embedded methods, PS-$l2$AROM and SVM.RFE.

Conclusion and Perspectives

The curse of dimensionality arises when analyzing data in high-dimensional spaces and results in weak predictive models with a limited generality, but also it is one of the main sources of feature selection instability. Feature selection is a solution for such problems. It is an important field of data mining and data knowledge discovery from many application domains. The development of stable feature selection algorithms is drawing increasing attention in various domains due to the importance of stability as an optimal feature selection criterion. Many feature selection algorithms have been proposed with the main objective of improving the predictive performance of learning algorithms. In this thesis, we focus also on the stability of feature selection and contribute to the study of stable feature selection through proposing algorithms and extensive empirical evaluation of the proposed methods. We propose new directions and ideas on providing stable feature selection algorithms.

We study the small size problem and propose feature selection methods that take into account this data specificity. Based on this concept, we derive three methods that are based on instance learning, one filter and two hybrid algorithms. These algorithms take advantage of the small sample size to allow choosing only a few subsets of features to be combined or analyzed. Small sample size makes this process feasible with acceptable running time. An instance based filter is first use to reduce the high dimensionality of data to few subsets of features which number corresponds to the data sample size. It selects relevant features by a simple combination scheme which calculates features' frequency of appearance in the different candidate subsets. As other alternatives, two wrapper approaches are proposed to integrate the performance of a predictive algorithm as an evaluation mechanism of the

candidate subsets obtained after the filter step. One of them is based on sequential backward search and the other is based on cooperative search of the best feature subset. The filter and the hybrid approach using a combination scheme based on a consensus decision making yield the best classification accuracy and stability of feature selection. These encouraging results guide us naturally to think of ensemble methods which precisely rely on the combination of multiple algorithms results to end with a unique solution.

Thus, in the second contribution of this thesis we study ensemble methods as a feature selection mechanism. We first proceed to a comparative study between different aggregation levels of ensemble feature selection, classifier and selector levels. We show the efficiency of ensemble feature selection in improving the classification performance. Then, we focus on ensemble selector aggregation level by proposing a robust feature aggregation technique to combine the results of different feature subsets. The interest of this approach is that it aims at providing a unique and stable feature selection without ignoring the predictive accuracy aspect. First, an ensemble of different feature subsets are obtained by function or data perturbation. After this step, a multiple classifier system is trained on each of the projections of the resulting feature subsets on the training data. The corresponding classification performances are used to measure the reliability of selected features and guide the final selection. Predictive performance and stability of the proposed method is compared to several existing ensemble feature selection aggregation schemes and the proposed method has shown to improve both evaluation criteria. The proposed method has also shown lower but close predictive performance compared to ensemble classifier method known to be a powerful mechanism for that purpose.

In our third contribution, in order to obtain robust feature selection results, we incorporated prior knowledge about some dimensions known to be more relevant, as a means of directing the feature selection process. We proposed new prior knowledge based extensions of three well known feature selection techniques. We proposed also a robust embedded feature selection method based on prior knowledge. This method makes use of a partial supervision on some features assumed a priori to be more relevant. Iteratively we make use of the initial prior knowledge and the previously selected features to expand a subset of highly relevant features in a pre-processing phase of feature selection. The proposed approaches have shown a positive effect especially on stability of feature selection, for high

dimensional small sample size data, compared to existing methods.

We have adopted stability and classification performance as main evaluation metrics along with cardinality of selected feature subsets, running time, statistical test and convergence behaviour of feature selection algorithm. Cross validation was essentially used as a validation protocol, while bootstrapping was used as a mechanism to create diverse selectors in the ensemble feature selection methods.

Experiments conducted on several microarray data sets have shown the effectiveness of our proposed contributions to handle the negative impact of the increasing ratio between the number of features and the sample size in such data sets, both on classification performance of the predictive models and on stability of feature selection.

The studies investigated in this thesis open new directions for future research. Different combination techniques could be applied with the instance based methods. Instance learning will be a new mechanism which can be used to produce relevant and diverse feature subsets. A combination of instance based method proposed in Chapter 1 and the robust aggregation method proposed in Chapter 2 could further improve stability and predictive performance. This topic is interesting to investigate.

Also, the comparative study on ensemble feature selection methods and the proposed robust aggregation technique could be extended to other feature selection methods. Studying the relationship between the baseline algorithm used for the creation of the selector ensemble and the ensemble aggregation mechanism would be interesting to further improve stability of ensemble feature selection methods.

A future direction concerning the prior knowledge based feature selection would be the investigation of our proposed semi supervised method as an optimization problem to maximize the two evaluation criteria, classification performance and stability, while integrating background knowledge about features relevancy. A multi-objective optimization problem is an interesting research area.

Bibliography

Abeel, T., T. Helleputte, Y. Van de Peer, P. Dupont, and Y. Saeys (2010). Robust biomarker identification for cancer diagnosis with ensemble feature selection methods. *Bioinformatics 26*(3), 392–398.

Aha, D. W. and B. R. L. (1996). *A Comparative Evaluation of Sequential Feature Selection Algorithms*, Volume 112.

Alizadeh, A. A., M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, R. Levy, W. Wilson, M. R. Grever, J. C. Byrd, D. Botstein, P. O. Brown, and L. M. Staudt (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature 403*(6769), 503–511.

Blum, A. L. and P. Langley (1997). Selection of relevant features and examples in machine learning. *Artificial Intelligence 97*, 245–271.

Bonett, D. G. and T. A. Wright (2000). Sample size requrements for estimating pearson, kendall and spearman correlations. *Psychometrika 65*(1), 23–28.

Bouckaert, R. R., E. Frank, M. Hall, R. Kirkby, P. Reutemann, A. Seewald, and D. Scuse (2009, June). *Weka manual (3.7.1)*.

Breiman, L., F. J. O. R. and C. Stone (1984). *Classification and Regression Trees*. Wadsworth Publishing Company.

Breiman, L. (1996). Bagging predictors. *Machine Learning 24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine Learning 45*(1), 5–32.

Carterette, B. (2009). On rank correlation and the distance between rankings. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, New York, NY, USA, pp. 436–443. ACM.

Chan, D., S. M. Bridges, and S. C. Burgess (2008). *An ensemble method for identifying robust features for biomarker discovery*. Chapman and Hall/CRC Press.

Costa, E. P., A. C. Lorena, Carvalho, and A. A. Freitas (2007). A review of performance evaluation measures for hierarchical classifiers. In *2007 AAAI Workshop, Vancouver*. AAAI Press.

Cover, T. and P. Hart (1967). Nearest neighbor pattern classification. *13*, 21– 27.

Dash, M. and H. Liu (1997). Feature selection for classification. *Intelligent Data Analysis 1*, 131–156.

Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation 10*, 1895–1923.

Dietterich, T. G. (2000). Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, London, UK, UK, pp. 1–15. Springer-Verlag.

Dinu, L. P. and F. Manea (2006, August). An efficient approach for the rank aggregation problem. *Theor. Comput. Sci. 359*(1), 455–461.

Duda, R. O., P. E. Hart, and D. G. Stork (2001). *Pattern Classification*. Wiley.

Dyrskjot, L., T. Thykjaer, M. Kruhoffer, J. L. Jensen, N. Marcussen, S. Hamilton-Dutoit, H. Wolf, and T. F. Orntoft (2003). Identifying distinct classes of bladder carcinoma using microarrays. *Nat Genetics 33*, 90–96.

Eveitt, B. (1977). The analysis of contingency tables. *Chapman and Hall*.

Fan, J. and Y. Fan (2008). High dimensional classification using features annealed independence rules. *Annals of statistics 36*(6), 2605–2637.

Fix, E. and J. Hodges (1951). *Discriminatory Analysis: Nonparametric Discrimination: Consistency Properties*. USAF School of Aviation Medicine.

Freund, Y. and R. E. Schapire (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. Syst. Sci. 55*(1), 119–139.

Garcia, M. A. and D. Puig (2003). Robust aggregation of expert opinions based on conflict analysis and resolution. In *CAEPIA*, Lecture Notes in Computer Science, pp. 488–497. Springer.

Golub, T. R., D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, and C. D. Bloomfield (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science 286*, 531–537.

Gordon, G., R. Jensen, L. Hsiao, S. Gullans, J. Blumenstock, S. Ramaswamy, W. Richards, D. Sugarbaker, and R. Bueno (2002). Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma. *Cancer Research 62*, 4963–4967.

Gosset, W. S. (1908). The probable error of a mean. *Biometrika* (1), 1–25.

Guan, P., D. Huang, M. He, and B. Zhou (2009). Lung cancer gene expression database analysis incorporating prior knowledge with support vector machine-based classification method. *Journal of Experimental and Clinical Cancer Research 28*, 103.

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

Guyon, I., J. Weston, S. Barnhill, and V. Vapnik (2002). Gene selection for cancer classification using support vector machines. *Machine Learning 46*, 389–422.

Hall, M. A. (2000). Correlation-based feature selection for discrete and numeric class machine learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 359–366. Morgan Kaufmann.

Hansen, L. and P. Salamon (1990). Neural network ensembles. *IEEE Transactions on Pattern Analysis and Machine Intelligence 12*(10), 993–1001.

He, Z. and W. Yu (2010). Review article: Stable feature selection for biomarker discovery. *Computational Biology and Chemistry 34*(4), 215–225.

Helleputte, T. and P. Dupont (2009a). Feature selection by transfer learning with linear regularized models. In *Proceedings of the of the European Conference on Machine Learning and Knowledge Discovery in Databases*, Berlin, pp. 533–547. Springer.

Helleputte, T. and P. Dupont (2009b). Partially supervised feature selection with regularized linear models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 409–416.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE Trans. Pattern Anal. Mach. Intell. 20*(8), 832–844.

Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems*. University of Michigan Press.

Hu, Q., W. Pan, Y. Song, and D. Yu (2012). Large-margin feature selection for monotonic classification. *Knowledge-Based Systems 31*, 8–18.

Huang, J., Y. Cai, and X. Xu (2007). A hybrid genetic algorithm for feature selection wrapper based on mutual information. *Pattern Recognition Letters 28*(13), 1825–1844.

Jolliffe, I. (1986). *Principal Component Analysis*. Springer Verlag.

Kalousis, A., J. Prados, and M. Hilario (2007). Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowl. Inf. Syst. 12*(1), 95–116.

Kalousis, A., J. Prados, J.-C. Sanchez, L. Allard, and M. Hilario (2004). Distilling classification models from cross validation runs: An application to mass spectrometry. In *ICTAI*, pp. 113–119. IEEE Computer Society.

Kira, K. and L. Rendell (1992). A practical approach to feature selection. In D. Sleeman and P. Edwards (Eds.), *International Conference on Machine Learning*, pp. 368–377.

Kohane, I. S., A. T. Kho, and A. J. Butte (2003). *Microarrays for an integrative genomics*. Cambridge, MA: MIT Press.

Kohavi, R. (1995). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, pp. 1137–1143. Morgan Kaufmann Publishers Inc.

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

Kolde, R., S. Laur, P. Adler, and J. Vilo (2012). Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics 28*(4), 573–580.

Kononenko, I. (1994). Estimating attributes: Analysis and extensions of relief. In *Proceedings of the European Conference on Machine Learning on Machine Learning*, pp. 171–182. Springer-Verlag New York, Inc.

Kumar, R. and S. Vassilvitskii (2010). Generalized distances between rankings. In *Proceedings of the 19th international conference on World wide web*, WWW '10, New York, NY, USA, pp. 571–580. ACM.

Kuncheva, L. (2007). A stability index for feature selection. In *Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, Innsbruck, Austria, pp. 390–395.

Kuncheva, L., J. Bezdek, and R. Duin (2001). Decision templates for multiple classifier fusion: An experimental comparison. *Pattern Recognition 34*, 299–314.

Lal, T., O. Chapelle, J. Weston, and A. Elisseeff (2006). Embedded methods. *207*, 137–165.

Langley, P. (1994). Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance*, pp. 140–144. AAAI Press.

Lee, S.-I., V. Chatalbashev, D. Vickrey, and D. Koller (2007). Learning a meta-level prior for feature relevance from multiple related tasks. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, New York, NY, USA, pp. 489–496. ACM.

Li, Y. and B. L. Lu (2009). Feature selection based on loss-margin of nearest neighbor classification. *Pattern Recognition 42*(9), 1914–1921.

Littlestone, N. and M. Warmuth (1994). Weighted majority algorithm. *Information and Computation 108*, 212–261.

Loscalzo, S., L. Yu, and C. H. Q. Ding (2009). Consensus group stable feature selection. In *KDD*, pp. 567–576. ACM.

Mitchell, L., T. Sloan, M. Mewissen, P. Ghazal, T. Forster, M. Piotrowski, and A. Trew (2014). Parallel classification and feature selection in microarray data using sprint. *Concurrency and Computation: Practice and Experience 26*(4), 854–865.

Okun, O. (2011). *Feature Selection and Ensemble Methods for Bioinformatics: Algorithmic Classification and Implementations*.

Opitz, D. W. (1999). Feature selection for ensembles. In *In Proceedings of 16th National Conference on Artificial Intelligence (AAAI*, pp. 379–384. Press.

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

Pihur, V., S. Datta, and S. Datta (2009). RankAggreg, an R package for weighted rank aggregation. *BMC Bioinformatics 10*(1), 62+.

Pomeroy, S. L., P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature 415*(6870), 436–442.

Quinlan, J. R. (1986). Induction of decision trees. *MACH. LEARN 1*, 81–106.

Quinlan, J. R. (1993). *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc.

Robnik, S. and I. Kononenko (2003). Theoretical and empirical analysis of relieff and rrelieff. *Machine Learning 53*(1-2), 23–69.

Saeys, Y., T. Abeel, and Y. Peer (2008). Robust feature selection using ensemble feature selection techniques. In *Proceedings of the European conference on Machine Learning and Knowledge Discovery in Databases - Part II*, ECML PKDD '08, Berlin, Heidelberg, pp. 313–325. Springer-Verlag.

Saeys, Y., I. Inza, and P. Larranaga (2007). A review of feature selection techniques in bioinformatics. *Bioinformatics 23*, 2507–2517.

Schapire, R. E. (1990). The strength of weak learnability. *Machine Learning 5*, 197–227.

Schowe, B. and K. Morik (2011). Fast-ensembles of minimum redundancy feature selection. In *Ensembles in Machine Learning Applications : Studies in Computational Intelligence*, Volume 373, pp. 75–95.

Shipp, M. A., K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, and D. S. Neuberg (2002). Diffuse large b(cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine 9*, 68–74.

Singh, D., P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, and J. P. Richie (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell 1*(2), 203–209.

Skalak, D. B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. In *Machine Learning: Proceedings of the Eleventh International Conference*, pp. 293–301. Morgan Kaufmann.

Spearman, C. (1987). The proof and measurement of association between two things. by c. spearman, 1904. *The American journal of psychology 100*(3-4), 441–471.

Stone, M. (1974). *Journal of the Royal Statistical Society. Series B (Methodological) 36*(2), 111–147.

Sun, Y., S. Todorovic, and S. Goodison (2010). Local learning based feature selection for high dimensional data analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence (TPAMI) 32*, 1610–1626.

Taskar, B., M. F. Wong, and D. Koller (2003). Learning on the test data: Leveraging unseen features. In *Proc. ICML*.

Tolosi, L. and T. Lengauer (2011). Classification with correlated features: unreliability of feature ranking and solutions. *Bioinformatics 27*(14), 1986–1994.

Troyanskaya, O. G., M. Cantor, G. Sherlock, P. O. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman (2001). Missing value estimation methods for dna microarrays. *Bioinformatics 17*(6), 520–525.

Tsymbal, A., M. Pechenizkiy, and P. Cunningham (2005). Sequential genetic search for ensemble feature selection. In *Proceedings of the 19th international joint conference on Artificial intelligence*, IJCAI'05, San Francisco, CA, USA, pp. 877–882. Morgan Kaufmann Publishers Inc.

Vafaie, H. and K. D. Jong (1992). Genetic algorithms as a tool for feature selection in machine learning. In *Proceedings 4th International Conference on Tools with Artificial Intelligence*, pp. 200–204. Society Press.

van der Maaten, L., E. O. Postma, and H. J. van den Herik (2008). Dimensionality reduction: A comparative review.

van 't Veer, L. J., H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend (2002, January). Gene expression profiling predicts clinical outcome of breast cancer. *Nature 415*(6871), 530–536.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Springer-Verlag New York, Inc.

Weston, J., A. Elisseeff, B. Schlkopf, and P. Kaelbling (2003). Use of the zero-norm with linear models and kernel methods. *Journal of Machine Learning Research 3*, 1439–1461.

Wu, X., V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg (2007). Top 10 algorithms in data mining. *Knowledge Information Systems 14*(1), 1–37.

Wyse, N., R. Dubes, and A. Jain (1980). A critical evaluation of intrinsic dimensionality algorithms a critical evaluation of intrinsic dimensionality algorithms. *Pattern Recognition in Practice*, 415425.

Xie, J., W. Xie, C. Wang, and X. Gao (2010). A novel hybrid feature selection method based on ifsffs and svm for the diagnosis of erythemato-squamous diseases. *Journal of Machine Learning Research - Proceedings Track*, 142–151.

Zhao, Z., J. Wang, H. Liu, J. Ye, and Y. Chang (2008). Identifying biologically relevant genes via multiple heterogeneous data sources. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, New York, NY, USA, pp. 839–847. ACM.

List of publications

The methods proposed in this thesis resulted in some publications. Three papers are submitted to international journals:

- A paper entitled "A Hybrid Feature Selection Method Based On Instance Learning And Cooperative Subset Search" is submitted to the Pattern Recognition Letters journal.
- A paper entitled "Ensemble feature selection for high dimensional data: A new method and a comparative study" is submitted to the journal of Advances in Data Analysis and Classification.
- A paper entitled "Stable instance based feature selection for high dimensional data using redundancy elimination and subsets aggregation" is submitted to the journal of Intelligent Data Analysis.

The publications related to this thesis are cited below.

Ben Brahim, A., Bouaguel, W. and Limam, M. (2014). Combining Feature Selection and Data Classification Using Ensemble Approaches: Application to Cancer Diagnosis and Credit Scoring. *Case Studies in Intelligent Computing: Achievements and Trends*, CRC Press, Taylor and Francis, New York, USA , 517 -534.

Ben Brahim, A. and Limam, M.(2014). A stable instance based filter for feature selection in

small sample size data sets. *Proceedings of the* 10^{th} *International Conference on Advanced Data Mining and Applications, Lecture Notes in Computer Science*, 8933, Springer, Guilin, China.

Ben Brahim, A. and Limam, M. (2014). New Prior Knowledge Based Extensions for Stable Feature Selection. *Proceedings of the* 6^{th} *International Conference of Software Computing and Pattern Recognition*, IEEE, Tunis, Tunisia.

Ben Brahim, A. Bouaguel, W. and Limam, M.(2013). Feature Selection Aggregation Versus Classifiers Aggregation for Several Data Dimensionalities, *International Conference on Control, Engineering and Information Technology (CEIT13), Economics and Strategic Management of Business Process,* vol.1, 10 - 15.

Bouaguel, W., Ben Brahim, A. and Limam, M.(2013). Feature Selection By Rank Aggregation and Genetic Algorithms, *Proceedings of the* 5^{th} *International Conference on Knowledge Discovery and Information Retrieval*, IEEE, Vilamoura, Algarve, Portugal.

Ben Brahim, A. and Limam, M.(2013). Robust Ensemble Feature Selection for High Dimensional Data Sets, *Proceedings of the* 11^{th} *International Conference on High Performance Computing and Simulation*, IEEE, Helsinki, Finland.

Ben Brahim, A., Kalousis, A. and Limam, M.(2012). Feature Selection by Incorporating Prior Knowledge for High Dimensional Data Classification, *Proceedings of the the* 3^{rd} *Meeting on Statistics and Data Mining*. Hammamet, Tunisia.

Appendix A

Ensemble Feature Selection Results

A.1 Introduction

This appendix presents detailed results of the application of several ensemble feature selection methods using the function perturbation, then the data perturbation settings. These results are related to the robust ensemble feature selection based on multiple classifiers performance, second part of Chapter 3.

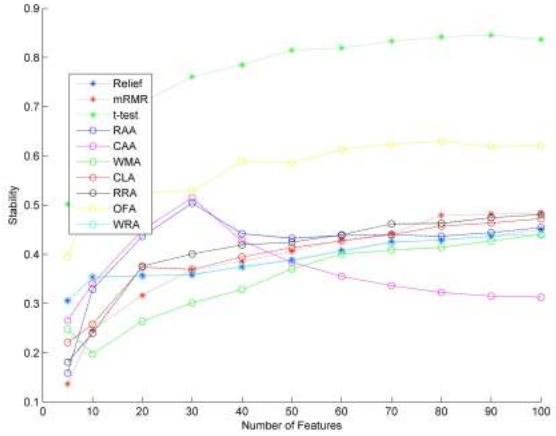
A.2 Classification and stability performances

For DLBCL data set and based on Table (A.1), with data perturbation setting using Relief as baseline algorithm, ECA and RAA give the two best MCE for all data perturbation settings. WMA gives the best MCE for the function perturbation setting and a good MCE for t-test data perturbation. We notice that for all settings MCE decreases when the number of features increases. It may reach its optimum value than increases again.

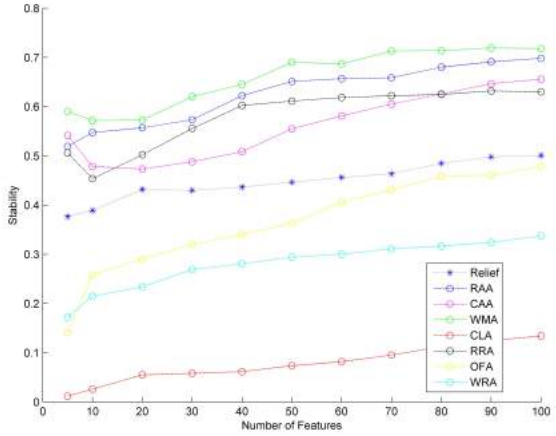
Table A.1: Classification error rates of ensemble methods for DLBCL data set.

N	Rel	mrmr	ttest	ECA	RAA	WMA	CLA	CAA	RRA	OFA	WRA
10	0.182	0.143	0.351	0.143	0.143	0.221	0.13	0.208	0.169	0.234	0.195
20	0.195	0.130	0.260	0.104	0.117	0.143	0.13	0.091	0.104	0.208	0.195
30	0.104	0.130	0.221	0.104	0.169	0.208	0.195	0.104	0.104	0.182	0.078
40	0.065	0.143	0.221	0.117	0.143	0.13	0.13	0.078	0.104	0.143	0.078
50	0.065	0.104	0.234	0.117	0.117	0.117	0.117	0.052	0.104	0.104	0.065
60	0.052	0.078	0.182	0.052	0.078	0.091	0.091	0.078	0.052	0.078	0.052
70	0.052	0.078	0.169	0.04	0.078	0.078	0.052	0.052	0.052	0.065	0.039
80	0.039	0.091	0.156	0.052	0.065	0.078	0.052	0.052	0.052	0.052	0.026
90	0.026	0.091	0.156	0.065	0.039	0.065	0.065	0.052	0.052	0.039	0.026
100	0.026	0.065	0.143	0.052	0.039	0.065	0.052	0.052	0.052	0.026	0.026
10	0.199			0.065	0.182	0.182	0.221	0.208	0.26	0.208	0.156
20	0.156			0.078	0.117	0.117	0.091	0.169	0.156	0.143	0.156
30	0.156			0.065	0.091	0.078	0.117	0.143	0.091	0.117	0.13
40	0.104			0.078	0.078	0.065	0.091	0.195	0.104	0.052	0.078
50	0.104			0.052	0.052	0.052	0.091	0.104	0.078	0.065	0.104
60	0.091			0.065	0.065	0.039	0.078	0.091	0.065	0.091	0.065
70	0.065			0.052	0.052	0.052	0.065	0.104	0.065	0.078	0.078
80	0.065			0.052	0.052	0.052	0.052	0.091	0.065	0.078	0.065
90	0.065			0.052	0.052	0.052	0.052	0.078	0.078	0.078	0.065
100	0.078			0.052	0.052	0.052	0.052	0.091	0.078	0.078	0.065
10		0.116		0.077	0.129	0.324	0.194	0.233	0.168	0.168	0.194
20		0.116		0.077	0.181	0.142	0.194	0.220	0.116	0.129	0.129
30		0.142		0.039	0.103	0.090	0.129	0.207	0.129	0.207	0.129
40		0.077		0.039	0.116	0.090	0.077	0.181	0.194	0.194	0.103
50		0.077		0.064	0.077	0.064	0.064	0.103	0.116	0.142	0.051
60		0.077		0.064	0.051	0.064	0.077	0.116	0.116	0.116	0.077
70		0.077		0.039	0.051	0.051	0.051	0.155	0.090	0.051	0.077
80		0.051		0.026	0.039	0.039	0.039	0.142	0.077	0.051	0.051
90		0.064		0.026	0.064	0.051	0.064	0.155	0.064	0.051	0.051
100		0.064		0.026	0.064	0.051	0.064	0.142	0.064	0.051	0.039
10			0.194	0.103	0.259	0.376	0.376	0.259	0.324	0.376	0.311
20			0.220	0.077	0.272	0.285	0.298	0.155	0.272	0.272	0.142
30			0.233	0.103	0.207	0.246	0.194	0.116	0.207	0.259	0.142
40			0.207	0.090	0.194	0.116	0.168	0.103	0.129	0.181	0.155
50			0.155	0.090	0.155	0.168	0.194	0.155	0.129	0.155	0.116
60			0.168	0.077	0.129	0.142	0.116	0.181	0.116	0.155	0.090
70			0.129	0.077	0.116	0.168	0.116	0.181	0.116	0.142	0.064
80			0.155	0.051	0.155	0.103	0.116	0.155	0.090	0.103	0.051
90			0.181	0.051	0.103	0.077	0.103	0.168	0.155	0.090	0.039
100			0.168	0.051	0.064	0.090	0.090	0.181	0.116	0.077	0.039

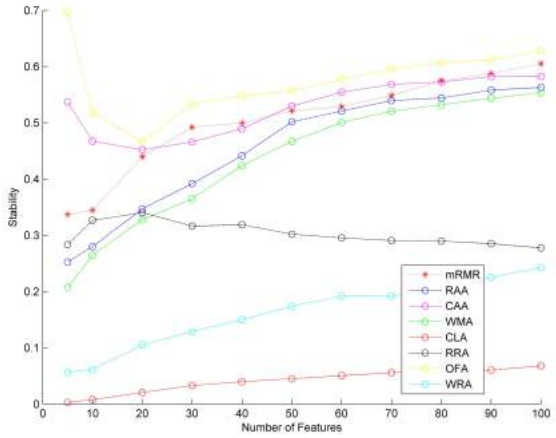
For Bladder cancer data set, Table (A.2) shows that RAA and WMA are outperforming for function perturbation setting. RAA is also among the best methods for data perturbation settings with mRMR and t-test as baselines.



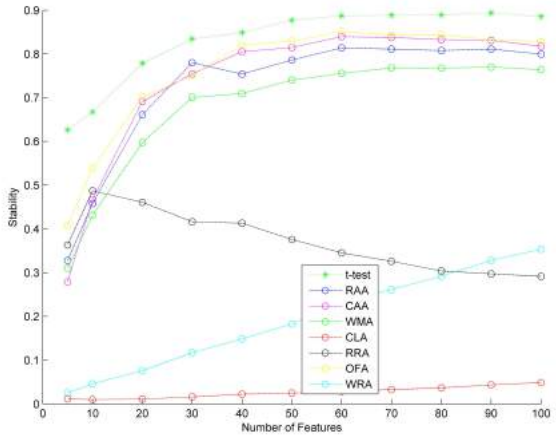
(a) Function perturbation



(b) Data perturbation - Relief



(c) Data perturbation - mRMR



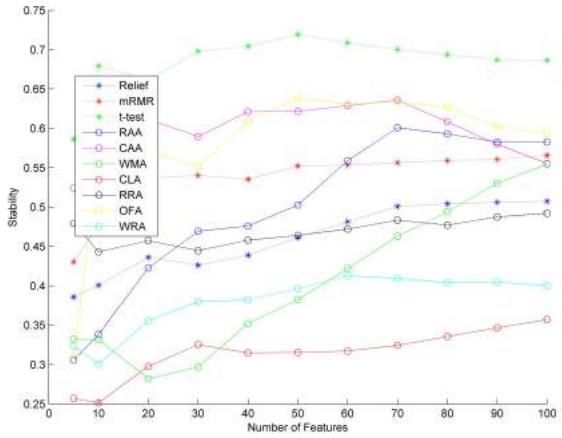
(d) Data perturbation - t-test

Figure A.1: Stability of ensemble methods for Bladder cancer data set.

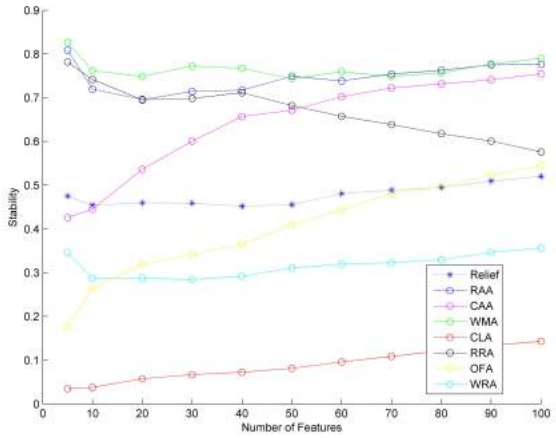
Table A.2: Classification error rates of ensemble methods for Bladder data set.

N	Rel	mrmr	ttest	ECA	RAA	WMA	CLA	CAA	RRA	OFA	WRA
10	0.225	0.193	0.161	0.161	0.096	0.161	0.161	0.225	0.161	0.161	0.225
20	0.258	0.193	0.161	0.193	0.129	0.161	0.193	0.193	0.129	0.129	0.258
30	0.161	0.161	0.129	0.129	0.064	0.064	0.193	0.096	0.129	0.096	0.193
40	0.161	0.225	0.161	0.193	0.064	0.096	0.161	0.096	0.096	0.129	0.161
50	0.129	0.161	0.161	0.129	0.096	0.064	0.096	0.096	0.064	0.129	0.129
60	0.096	0.193	0.161	0.129	0.096	0.096	0.129	0.096	0.096	0.129	0.096
70	0.064	0.096	0.161	0.096	0.129	0.096	0.096	0.064	0.096	0.096	0.064
80	0.129	0.096	0.161	0.129	0.096	0.096	0.096	0.064	0.064	0.129	0.129
90	0.096	0.096	0.096	0.064	0.096	0.064	0.096	0.096	0.096	0.096	0.096
100	0.096	0.096	0.096	0.096	0.096	0.064	0.096	0.096	0.096	0.096	0.096
10	0.387			0.225	0.322	0.290	0.483	0.387	0.419	0.161	0.258
20	0.322			0.258	0.290	0.193	0.290	0.483	0.387	0.193	0.193
30	0.290			0.258	0.225	0.193	0.258	0.354	0.096	0.161	0.290
40	0.225			0.225	0.129	0.193	0.193	0.322	0.225	0.193	0.225
50	0.225			0.161	0.096	0.129	0.161	0.322	0.193	0.193	0.161
60	0.225			0.161	0.161	0.161	0.161	0.290	0.161	0.225	0.161
70	0.193			0.096	0.193	0.161	0.129	0.258	0.161	0.193	0.129
80	0.161			0.096	0.193	0.161	0.193	0.161	0.193	0.129	0.129
90	0.161			0.096	0.161	0.193	0.161	0.096	0.161	0.161	0.129
100	0.193			0.096	0.161	0.193	0.193	0.064	0.193	0.096	0.096
10		0.129		0.161	0.064	0.161	0.064	0.193	0.161	0.129	0.225
20		0.129		0.129	0.129	0.161	0.096	0.258	0.193	0.322	0.193
30		0.129		0.129	0.193	0.161	0.129	0.096	0.225	0.161	0.161
40		0.129		0.129	0.161	0.193	0.161	0.129	0.225	0.193	0.161
50		0.129		0.161	0.161	0.161	0.161	0.129	0.225	0.161	0.129
60		0.129		0.129	0.161	0.161	0.193	0.129	0.225	0.193	0.129
70		0.129		0.193	0.193	0.161	0.161	0.129	0.193	0.161	0.129
80		0.129		0.129	0.096	0.129	0.129	0.193	0.129	0.161	0.161
90		0.129		0.193	0.096	0.129	0.096	0.193	0.161	0.193	0.129
100		0.161		0.225	0.129	0.129	0.161	0.161	0.129	0.193	0.129
10			0.064	0.096	0.096	0.096	0.064	0.225	0.129	0.096	0.161
20			0.064	0.096	0.032	0.064	0.064	0.193	0.129	0.064	0.096
30			0.096	0.129	0.032	0.064	0.096	0.193	0.161	0.064	0.096
40			0.096	0.096	0.032	0.032	0.096	0.161	0.161	0.129	0.129
50			0.096	0.096	0.032	0.064	0.096	0.193	0.161	0.064	0.129
60			0.096	0.096	0.096	0.064	0.096	0.225	0.161	0.064	0.129
70			0.096	0.096	0.129	0.161	0.096	0.225	0.161	0.129	0.129
80			0.032	0.129	0.096	0.129	0.096	0.161	0.161	0.129	0.129
90			0.032	0.096	0.129	0.129	0.129	0.193	0.129	0.161	0.129
100			0.032	0.096	0.193	0.193	0.193	0.193	0.129	0.193	0.129

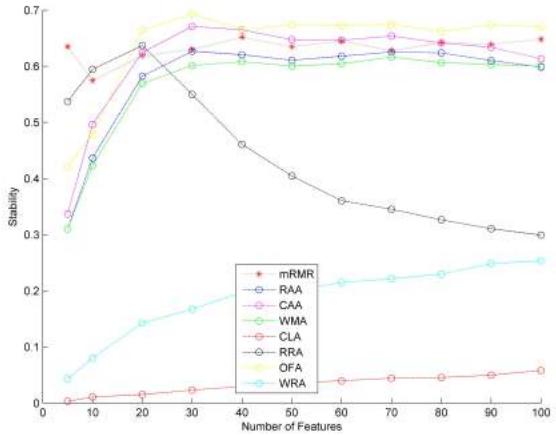
For Lymphoma cancer data set, Table (A.3) shows that ECA is outperforming for all settings, function and data perturbation. WMA is among the best methods for all data perturbation settings. RAA is also outperforming for data perturbation settings with Relief and mRMR as baselines.



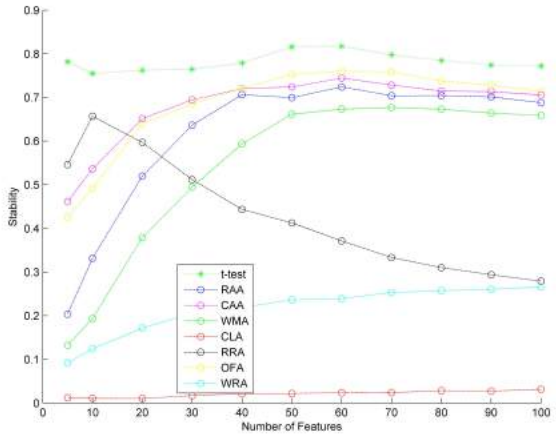
(a) Function perturbation



(b) Data perturbation - Relief



(c) Data perturbation - mRMR



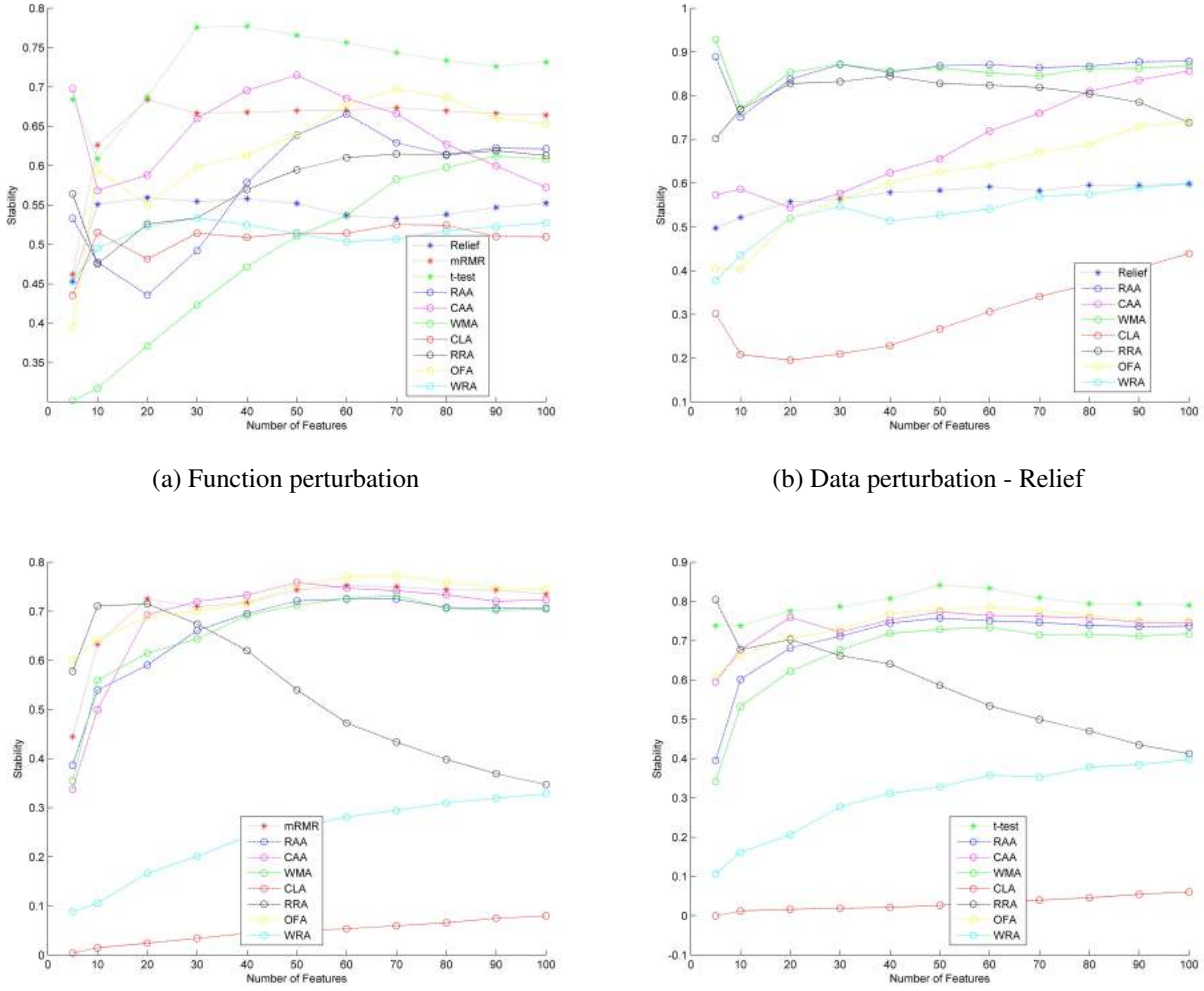
(d) Data perturbation - t-test

Figure A.2: Stability of ensemble methods for Lymphoma data set.

Table A.3: Classification error rates of ensemble methods for Lymphoma data set.

N	Rel	mrmr	ttest	ECA	RAA	WMA	CLA	CAA	RRA	OFA	WRA
10	0.266	0.066	0.155	0.088	0.088	0.044	0.044	0.155	0.044	0.133	0.222
20	0.133	0.022	0.022	0	0.088	0.022	0.022	0.133	0.044	0.022	0.266
30	0.066	0.044	0.044	0.022	0	0.044	0	0.066	0.044	0.022	0.155
40	0.022	0.044	0.066	0.022	0.044	0.044	0.022	0.044	0.044	0.022	0.155
50	0.022	0.044	0.044	0.022	0.044	0.044	0.022	0.066	0.022	0.044	0.222
60	0.044	0.022	0.044	0.022	0.044	0.044	0	0.044	0.022	0.044	0.222
70	0.044	0.044	0.044	0.044	0.044	0.044	0.022	0.066	0.022	0.044	0.200
80	0.088	0.044	0.022	0.044	0	0.022	0.022	0.088	0.022	0	0.155
90	0.088	0.044	0.022	0.044	0	0.044	0.022	0.088	0.022	0	0.133
100	0.066	0.022	0.022	0.022	0	0.044	0	0.066	0.044	0	0.133
10	0.177			0.088	0.133	0.133	0.222	0.377	0.177	0.288	0.377
20	0.177			0.066	0.044	0.066	0.111	0.288	0.133	0.288	0.244
30	0.133			0.066	0.066	0.088	0.066	0.311	0.088	0.133	0.222
40	0.155			0.044	0.066	0.066	0.044	0.266	0.044	0.244	0.244
50	0.177			0.066	0.066	0.066	0.088	0.177	0.044	0.111	0.177
60	0.111			0.066	0.088	0.066	0.111	0.177	0.088	0.155	0.155
70	0.155			0.066	0.111	0.088	0.111	0.222	0.088	0.066	0.111
80	0.111			0.088	0.088	0.088	0.111	0.111	0.088	0.044	0.111
90	0.111			0.111	0.066	0.088	0.111	0.111	0.111	0.111	0.111
100	0.111			0.111	0.066	0.088	0.088	0.111	0.133	0.133	0.111
10		0.044		0.066	0.088	0.066	0.111	0.288	0.088	0.177	0.088
20		0		0.022	0.088	0.066	0.133	0.2	0.044	0.133	0.066
30		0.066		0.044	0.088	0.066	0.088	0.155	0	0.022	0.044
40		0.088		0.044	0.044	0	0.044	0.111	0.066	0.044	0.066
50		0.066		0.044	0.022	0.044	0.022	0.111	0.044	0.022	0.022
60		0.022		0.022	0.044	0	0	0.111	0.066	0	0.044
70		0.022		0.022	0	0	0	0.088	0.066	0.022	0.044
80		0.022		0.022	0	0	0.022	0.111	0.044	0.022	0.044
90		0.044		0.022	0.022	0	0.022	0.111	0.066	0.022	0.022
100		0.066		0.022	0.022	0.022	0.044	0.111	0.044	0.022	0.044
10			0.066	0.088	0.177	0.088	0.133	0.4	0.088	0.111	0.111
20			0.066	0.044	0.111	0.066	0.044	0.266	0.044	0.066	0.133
30			0.044	0.044	0.066	0.044	0.066	0.266	0.088	0.066	0.111
40			0.066	0.044	0.044	0.044	0.066	0.266	0.066	0.066	0.177
50			0.066	0.044	0.022	0.044	0.044	0.244	0.044	0.066	0.111
60			0.066	0.044	0.044	0.066	0.044	0.222	0.066	0.066	0.133
70			0.066	0.044	0.044	0.066	0.044	0.266	0.066	0.066	0.155
80			0.044	0.044	0.044	0.044	0.044	0.333	0.044	0.066	0.066
90			0.044	0.044	0.044	0.044	0.044	0.266	0.044	0.044	0.088
100			0.044	0.044	0.044	0.044	0.044	0.266	0.044	0.044	0.066

Table (A.4) shows that for Prostate cancer data set also, ECA and WMA is giving very good MCE perfromances.



(a) Function perturbation

(b) Data perturbation - Relief

(c) Data perturbation - mRMR

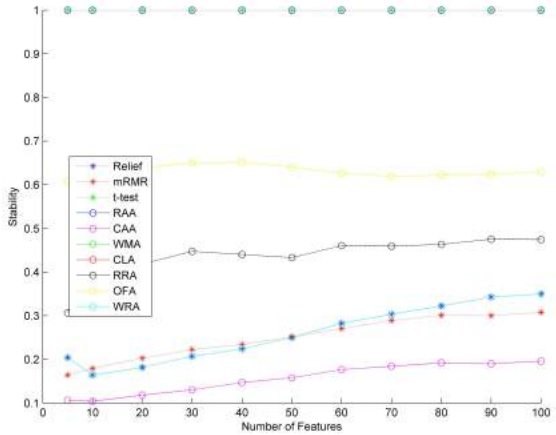
(d) Data perturbation - t-test

Figure A.3: Stability of ensemble methods for Prostate cancer data set.

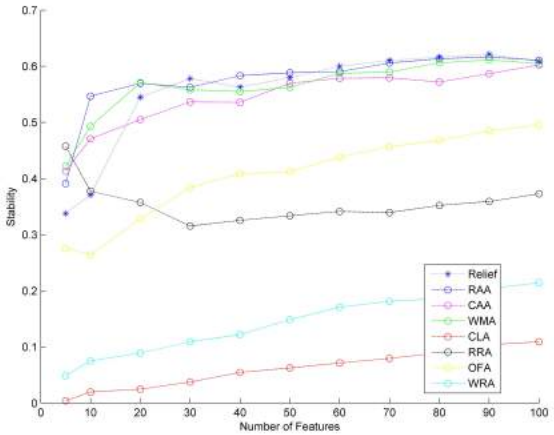
Table A.4: Classification error rates of ensemble methods for Prostate data set.

N	Rel	mrmr	ttest	ECA	RAA	WMA	CLA	CAA	RRA	OFA	WRA
10	0.303	0.098	0.196	0.156	0.254	0.166	0.166	0.186	0.166	0.176	0.274
20	0.225	0.078	0.156	0.127	0.176	0.117	0.127	0.147	0.088	0.196	0.186
30	0.186	0.078	0.156	0.098	0.117	0.088	0.098	0.137	0.117	0.147	0.225
40	0.176	0.078	0.117	0.088	0.107	0.088	0.137	0.176	0.176	0.156	0.196
50	0.147	0.078	0.078	0.049	0.058	0.088	0.098	0.205	0.176	0.107	0.156
60	0.107	0.078	0.078	0.068	0.078	0.098	0.078	0.166	0.166	0.088	0.166
70	0.137	0.068	0.078	0.078	0.068	0.107	0.078	0.166	0.166	0.127	0.147
80	0.107	0.088	0.088	0.078	0.068	0.098	0.088	0.137	0.117	0.107	0.117
90	0.107	0.098	0.088	0.078	0.088	0.098	0.098	0.127	0.127	0.088	0.137
100	0.117	0.098	0.098	0.078	0.078	0.078	0.098	0.117	0.117	0.078	0.127
10	0.264			0.186	0.264	0.186	0.245	0.274	0.264	0.343	0.352
20	0.333			0.107	0.196	0.107	0.196	0.245	0.196	0.254	0.225
30	0.245			0.098	0.147	0.088	0.166	0.196	0.166	0.166	0.235
40	0.254			0.088	0.088	0.049	0.107	0.147	0.117	0.166	0.196
50	0.245			0.078	0.098	0.039	0.117	0.176	0.107	0.117	0.186
60	0.225			0.078	0.088	0.078	0.098	0.176	0.127	0.137	0.147
70	0.225			0.068	0.107	0.068	0.107	0.166	0.107	0.137	0.166
80	0.235			0.078	0.098	0.078	0.098	0.156	0.107	0.156	0.156
90	0.264			0.088	0.098	0.088	0.098	0.137	0.088	0.137	0.117
100	0.245			0.078	0.107	0.088	0.088	0.107	0.088	0.137	0.107
10		0.088		0.098	0.196	0.147	0.196	0.372	0.147	0.147	0.284
20		0.147		0.127	0.117	0.147	0.117	0.254	0.127	0.098	0.098
30		0.156		0.107	0.137	0.098	0.166	0.205	0.117	0.127	0.117
40		0.147		0.098	0.137	0.117	0.137	0.196	0.107	0.156	0.137
50		0.137		0.098	0.127	0.098	0.107	0.176	0.137	0.117	0.107
60		0.127		0.098	0.117	0.068	0.088	0.196	0.127	0.117	0.147
70		0.137		0.098	0.117	0.117	0.117	0.235	0.127	0.117	0.137
80		0.147		0.068	0.098	0.107	0.098	0.205	0.127	0.088	0.127
90		0.137		0.088	0.088	0.088	0.107	0.205	0.117	0.078	0.107
100		0.127		0.078	0.088	0.078	0.088	0.205	0.117	0.078	0.117
10			0.137	0.088	0.186	0.156	0.205	0.333	0.215	0.196	0.166
20			0.117	0.068	0.156	0.156	0.176	0.215	0.156	0.147	0.156
30			0.117	0.068	0.117	0.147	0.137	0.215	0.156	0.127	0.098
40			0.137	0.078	0.107	0.137	0.127	0.196	0.127	0.147	0.107
50			0.117	0.068	0.078	0.078	0.088	0.235	0.117	0.117	0.088
60			0.107	0.068	0.107	0.098	0.107	0.235	0.117	0.107	0.098
70			0.127	0.078	0.098	0.098	0.127	0.205	0.058	0.098	0.107
80			0.117	0.098	0.117	0.098	0.127	0.147	0.068	0.078	0.088
90			0.117	0.088	0.137	0.098	0.127	0.156	0.068	0.098	0.088
100			0.098	0.098	0.098	0.078	0.117	0.166	0.068	0.117	0.098

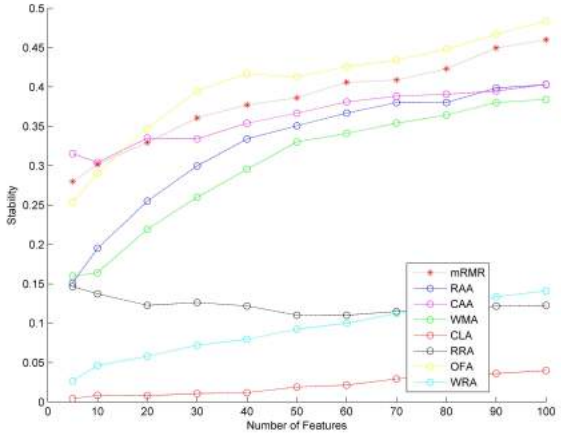
For Breast cancer data set, Table (A.5) shows that ECA is also outperforming for all settings. WMA performs well for function perturbation setting and data perturbation using mRMR and t-test baseline algorithms. RAA and OFA are among the best methods for two data perturbation settings out of three.



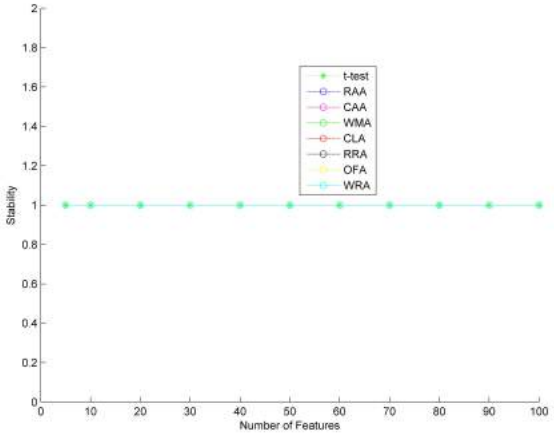
(a) Function perturbation



(b) Data perturbation - Relief



(c) Data perturbation - mRMR



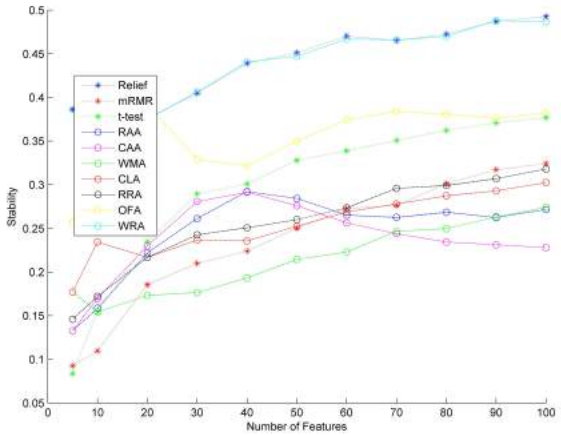
(d) Data perturbation - t-test

Figure A.4: Stability of ensemble methods for Breast cancer data set.

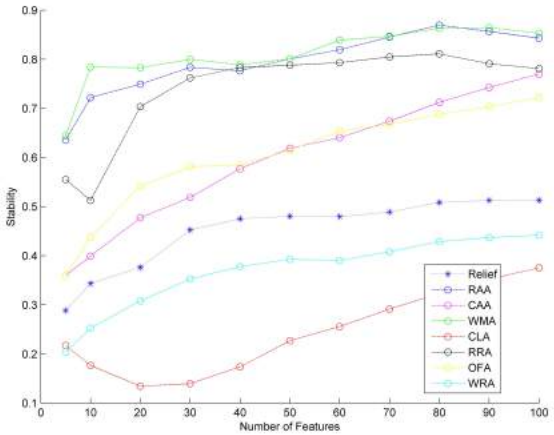
Table A.5: Classification error rates of ensemble methods for Breast cancer data set.

N	Rel	mrmr	ttest	ECA	RAA	WMA	CLA	CAA	RRA	OFA	WRA
10	0.422	0.319	0.515	0.371	0.515	0.350	0.422	0.494	0.494	0.505	0.422
20	0.443	0.329	0.515	0.391	0.515	0.350	0.443	0.463	0.463	0.505	0.443
30	0.484	0.381	0.515	0.422	0.515	0.443	0.484	0.463	0.463	0.494	0.484
40	0.453	0.340	0.515	0.381	0.515	0.371	0.453	0.525	0.525	0.494	0.453
50	0.463	0.350	0.515	0.453	0.515	0.360	0.463	0.463	0.463	0.494	0.463
60	0.484	0.340	0.515	0.402	0.515	0.371	0.484	0.484	0.484	0.515	0.484
70	0.443	0.319	0.515	0.371	0.515	0.319	0.443	0.546	0.546	0.505	0.443
80	0.402	0.299	0.515	0.371	0.515	0.309	0.402	0.515	0.515	0.515	0.402
90	0.412	0.319	0.515	0.391	0.515	0.319	0.412	0.505	0.505	0.474	0.412
100	0.381	0.340	0.515	0.391	0.515	0.360	0.381	0.463	0.463	0.463	0.381
10	0.422			0.453	0.556	0.463	0.577	0.525	0.422	0.494	0.412
20	0.381			0.443	0.505	0.453	0.443	0.494	0.494	0.443	0.453
30	0.402			0.463	0.484	0.391	0.433	0.525	0.360	0.453	0.433
40	0.422			0.402	0.474	0.494	0.474	0.433	0.381	0.463	0.402
50	0.474			0.433	0.402	0.433	0.463	0.505	0.484	0.474	0.381
60	0.443			0.453	0.422	0.463	0.453	0.505	0.463	0.443	0.422
70	0.453			0.412	0.433	0.463	0.453	0.505	0.484	0.412	0.463
80	0.422			0.422	0.453	0.433	0.422	0.494	0.463	0.422	0.484
90	0.453			0.453	0.463	0.474	0.443	0.463	0.391	0.463	0.505
100	0.443			0.412	0.443	0.433	0.453	0.422	0.412	0.412	0.525
10		0.340		0.34	0.35	0.299	0.36	0.288	0.34	0.381	0.34
20		0.412		0.278	0.371	0.340	0.371	0.360	0.299	0.288	0.402
30		0.381		0.299	0.329	0.329	0.371	0.402	0.329	0.329	0.381
40		0.329		0.288	0.329	0.309	0.391	0.391	0.340	0.299	0.371
50		0.329		0.319	0.319	0.309	0.319	0.371	0.309	0.288	0.371
60		0.309		0.319	0.299	0.299	0.340	0.340	0.329	0.340	0.371
70		0.319		0.319	0.299	0.257	0.299	0.329	0.309	0.329	0.340
80		0.309		0.278	0.319	0.288	0.319	0.319	0.360	0.340	0.340
90		0.278		0.299	0.288	0.268	0.309	0.350	0.340	0.288	0.360
100		0.268		0.309	0.288	0.257	0.288	0.350	0.319	0.288	0.350
10			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
20			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
30			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
40			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
50			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
60			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
70			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
80			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
90			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515
100			0.525	0.515	0.515	0.515	0.515	0.515	0.515	0.515	0.515

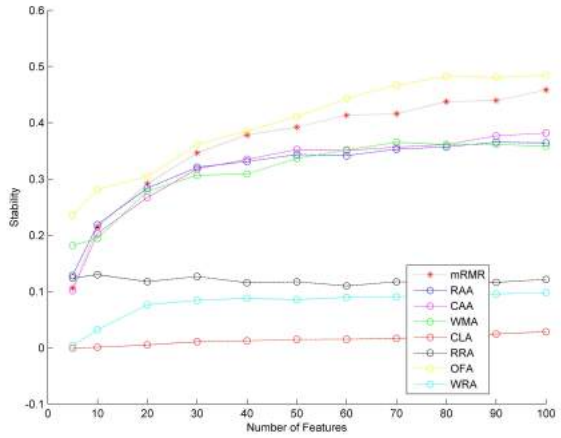
In Table (A.6) corresponding to CNS dataset results, RAA and OFA are the best for function and two data perturbation settings, based on mRMR and t-test baselines. OFA gives also good performance for Relief based data perturbation setting. ECA performs well for data perturbation based on Relief and t-test.



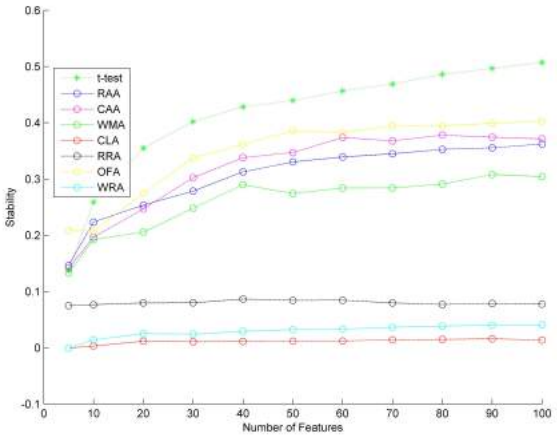
(a) Function perturbation



(b) Data perturbation - Relief



(c) Data perturbation - mRMR



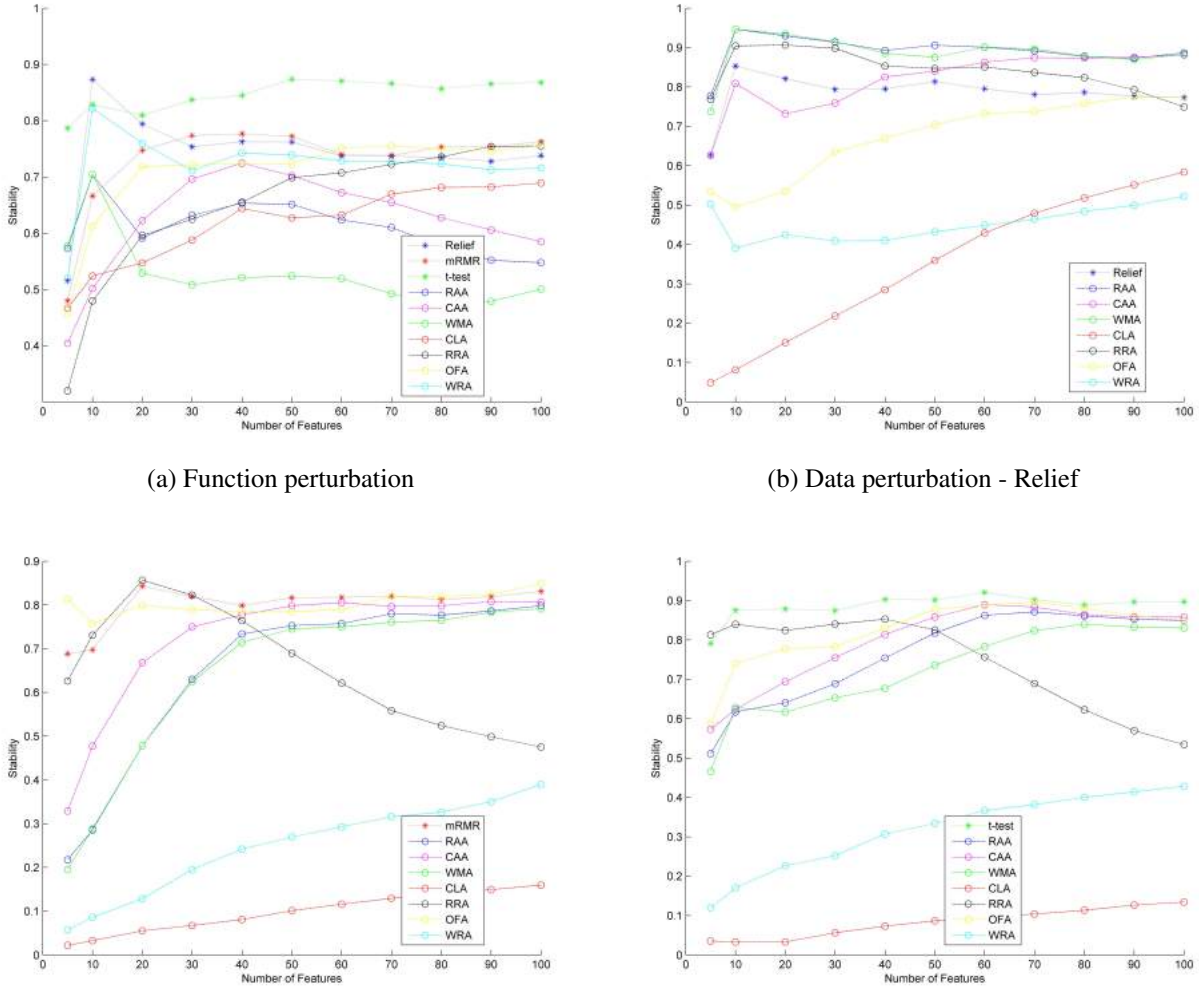
(d) Data perturbation - t-test

Figure A.5: Stability of ensemble methods for CNS data set.

Table A.6: Classification error rates of ensemble methods for CNS data set.

N	Rel	mrmr	ttest	ECA	RAA	WMA	CLA	CAA	RRA	OFA	WRA
10	0.433	0.400	0.450	0.433	0.516	0.533	0.516	0.500	0.483	0.400	0.433
20	0.566	0.466	0.483	0.516	0.450	0.433	0.416	0.483	0.450	0.316	0.566
30	0.583	0.466	0.500	0.550	0.450	0.416	0.416	0.550	0.533	0.366	0.583
40	0.400	0.483	0.466	0.416	0.333	0.366	0.483	0.483	0.500	0.316	0.400
50	0.500	0.433	0.400	0.400	0.400	0.350	0.383	0.516	0.533	0.416	0.483
60	0.383	0.433	0.416	0.383	0.366	0.333	0.400	0.433	0.450	0.366	0.383
70	0.433	0.366	0.516	0.433	0.283	0.366	0.433	0.500	0.450	0.450	0.433
80	0.400	0.300	0.466	0.400	0.350	0.366	0.366	0.433	0.466	0.433	0.433
90	0.433	0.300	0.383	0.350	0.333	0.416	0.383	0.433	0.433	0.383	0.433
100	0.333	0.350	0.366	0.350	0.350	0.433	0.416	0.483	0.400	0.350	0.333
10	0.433			0.383	0.450	0.383	0.500	0.433	0.433	0.366	0.516
20	0.483			0.433	0.500	0.366	0.483	0.416	0.550	0.566	0.416
30	0.466			0.350	0.433	0.333	0.350	0.500	0.483	0.333	0.416
40	0.483			0.400	0.416	0.333	0.400	0.416	0.416	0.350	0.450
50	0.400			0.433	0.500	0.283	0.483	0.450	0.433	0.366	0.400
60	0.416			0.350	0.366	0.316	0.466	0.400	0.433	0.316	0.383
70	0.400			0.316	0.416	0.350	0.416	0.416	0.466	0.383	0.416
80	0.416			0.283	0.350	0.316	0.383	0.416	0.450	0.350	0.350
90	0.400			0.283	0.333	0.300	0.316	0.400	0.316	0.350	0.383
100	0.383			0.266	0.283	0.316	0.300	0.400	0.350	0.350	0.416
10		0.266		0.566	0.566	0.583	0.516	0.450	0.483	0.516	0.583
20		0.316		0.416	0.433	0.466	0.450	0.483	0.416	0.466	0.500
30		0.300		0.433	0.400	0.433	0.433	0.450	0.500	0.416	0.400
40		0.300		0.416	0.400	0.383	0.466	0.483	0.500	0.483	0.450
50		0.366		0.400	0.383	0.333	0.400	0.516	0.383	0.416	0.400
60		0.450		0.400	0.350	0.316	0.433	0.433	0.366	0.466	0.383
70		0.400		0.416	0.383	0.416	0.383	0.416	0.350	0.400	0.383
80		0.383		0.383	0.366	0.416	0.416	0.400	0.333	0.383	0.383
90		0.400		0.383	0.366	0.433	0.383	0.400	0.333	0.416	0.383
100		0.316		0.383	0.383	0.383	0.383	0.383	0.350	0.383	0.383
10			0.516	0.500	0.533	0.516	0.483	0.483	0.483	0.533	0.566
20			0.516	0.350	0.466	0.450	0.400	0.400	0.500	0.416	0.366
30			0.500	0.316	0.366	0.433	0.416	0.433	0.383	0.350	0.400
40			0.500	0.333	0.283	0.416	0.350	0.466	0.400	0.316	0.416
50			0.533	0.333	0.333	0.316	0.300	0.466	0.433	0.366	0.400
60			0.400	0.350	0.316	0.316	0.333	0.483	0.333	0.416	0.433
70			0.416	0.366	0.300	0.250	0.316	0.483	0.383	0.366	0.433
80			0.400	0.366	0.300	0.300	0.350	0.400	0.400	0.350	0.466
90			0.450	0.350	0.350	0.333	0.366	0.366	0.366	0.366	0.366
100			0.450	0.366	0.333	0.350	0.350	0.316	0.316	0.350	0.416

For Lung cancer data set, Table (A.7) ECA is outperforming for all settings. RAA is among the best methods for function perturbation and t-test based data perturbation setting. WMA and OFA are among the best techniques each for two data perturbation settings.



(a) Function perturbation

(b) Data perturbation - Relief

(c) Data perturbation - mRMR

(d) Data perturbation - t-test

Figure A.6: Stability of ensemble methods for Lung cancer data set.

Table A.7: Classification error rates of ensemble methods for Lung cancer data set.

N	Rel	mrmr	ttest	ECA	RAA	WMA	CLA	CAA	RRA	OFA	WRA
10	0.060	0.033	0.044	0.005	0.055	0.044	0.055	0.022	0.016	0.022	0.060
20	0.060	0.016	0.033	0.011	0.027	0.011	0.022	0.016	0.016	0	0.055
30	0.016	0.016	0.027	0.011	0	0.011	0.016	0.016	0.016	0.011	0.022
40	0.022	0.016	0.049	0.011	0.005	0.005	0.011	0.016	0.011	0.005	0.022
50	0.011	0.016	0.038	0	0.022	0.016	0.011	0.005	0.011	0	0.016
60	0.011	0.011	0.038	0	0.011	0.005	0.005	0.005	0.011	0.005	0.005
70	0.011	0.005	0.022	0	0.011	0.005	0	0.005	0	0	0.011
80	0	0.005	0.022	0	0.005	0.005	0	0.005	0.005	0	0.005
90	0	0.005	0.022	0.005	0.005	0.005	0.005	0.005	0	0	0
100	0	0.005	0.016	0.005	0.005	0.005	0.005	0.005	0	0	0
10	0.044			0.060	0.060	0.105	0.060	0.099	0.055	0.044	0.082
20	0.016			0.038	0.077	0.038	0.060	0.038	0.066	0.022	0.044
30	0.016			0.016	0.033	0.016	0.038	0.044	0.033	0.011	0.044
40	0.022			0.022	0.005	0.005	0.016	0.033	0.016	0.016	0.016
50	0.022			0.011	0.016	0.011	0.016	0.033	0.011	0.005	0.011
60	0.016			0	0.011	0.005	0.016	0.033	0.016	0	0.011
70	0.016			0	0.011	0	0.016	0.027	0.011	0	0.011
80	0.016			0	0	0	0.005	0.011	0.005	0	0.011
90	0.016			0	0	0	0	0.016	0.005	0	0.016
100	0.016			0	0	0	0	0.005	0	0	0.011
10		0.011		0.022	0.055	0.071	0.055	0.071	0.038	0.011	0.077
20		0.016		0.016	0.044	0.016	0.044	0.033	0.016	0.027	0.060
30		0.011		0.016	0.022	0.016	0.027	0.033	0.011	0.016	0.016
40		0.011		0.016	0.005	0.016	0.011	0.016	0.011	0.005	0.022
50		0.027		0.005	0.011	0.011	0.011	0.016	0.016	0	0.005
60		0.027		0.005	0.011	0.005	0.016	0.022	0.022	0.005	0.005
70		0.027		0.005	0.005	0.005	0.011	0.016	0.016	0.005	0.011
80		0.027		0.005	0.005	0.005	0.005	0.016	0.016	0.005	0.005
90		0.027		0.005	0	0.005	0.005	0.011	0.016	0.005	0
100		0.022		0.005	0	0	0	0.005	0.011	0.005	0.005
10			0.027	0.027	0.044	0.105	0.049	0.038	0.033	0.044	0.342
20			0.033	0.011	0.022	0.038	0.060	0.044	0.027	0.044	0.044
30			0.044	0.016	0.022	0.033	0.027	0.038	0.027	0.033	0.016
40			0.055	0.011	0.027	0.022	0.027	0.038	0.027	0.049	0.027
50			0.049	0.011	0.027	0.022	0.016	0.033	0.016	0.027	0.022
60			0.044	0.011	0.011	0.038	0.016	0.038	0.016	0.022	0.016
70			0.044	0.011	0.027	0.011	0.022	0.033	0.011	0.016	0.011
80			0.038	0.022	0.016	0.022	0.022	0.027	0.011	0.016	0.011
90			0.038	0.016	0.016	0.011	0.016	0.033	0.022	0.016	0.011
100			0.038	0.016	0.011	0.022	0.011	0.033	0.022	0.011	0.016

Based on Figures (A.1) - (A.6), function perturbation ensemble methods yield modest stability performances which are smaller than t-test baseline method for all data sets. RAA and WMA give also good stability results for the three data perturbation settings. For the same setting, t-test as baseline, is slightly better than RAA, WMA, CAA and OFA. With the function perturbation setting, we notice that t-test clearly outperforms all ensemble methods and other baseline functions in terms of stability. However, in terms of classification performance, results show that t-test gives very poor or modest results using the same setting. It is to notice that CLA and WRA are giving poor stability results for all settings. Both techniques rely on feature ranking. The highest stability results for all data sets are obtained with data perturbation with Relief baseline algorithm. Moreover, gains increase with increasing feature subset sizes. However, for ECA we have often good classification performances. This is not good enough, since there is not a corresponding stability performance, as the objective of ECA is to aggregate classifier results built on different feature subsets obtained by applying the baseline algorithm on different data samples, and not to have a stable feature selection.

A.3 Conclusion

In this appendix, we have detailed classification and stability results of the ensemble feature selection methods discussed in Chapter 3. We see that stability behaviour is the same for all data sets. Concerning the predictive performance, the classification error decreases with the number of features increase. It reaches an optimum then become stable or increase. After analysing all experimental results, we conclude that ECA is the technique to use to get the best classification results. If feature selection stability is also important, RAA and WMA are the most efficient ensemble feature selection methods.

Résumé:

Le progrès technologique permet une évolution rapide du volume des données. Dans plusieurs domaines, cette tendance concerne surtout les dimensions des données. C'est le cas ou on trouve des milliers et des dizaines de milliers de variables avec un nombre d'instances beaucoup plus petit. Un tel cadre affecte les capacités d'apprentissage et de prévision des algorithmes d'apprentissage automatique. Ce phénomène est connu sous le nom de « fléau de la dimension ». La sélection de variables est une solution dans de telles situations. Cette thèse porte sur la classification et la sélection des variables de données à hautes dimensions. Elle est consacrée à la conception et l'application de méthodes efficaces de sélection de variables qui permettent l'estimation de modèles avec de bonnes performances de classification et avec une stabilité de la sélection. Cette thèse propose plusieurs moyens pour gérer le manque d'échantillons et la présence d'un très grand nombre de variables. L'apprentissage basé sur les instances, les méthodes d'ensembles et les méthodes de sélection de variables basées sur les connaissances à priori, constituent les principaux concepts proposés. Les méthodes de sélection développées améliorent la performance de classification et la stabilité.

Mots clés: Sélection de variables, haute dimensionnalité, stabilité, apprentissage basé sur les instances, méthodes d'ensemble, connaissances à priori.

Abstract:

The advanced technologies make amounts of data growing in a fast paced way. In many application fields, this trend concerns specially dimensions of the data. It is the case where features are about thousands and tens of thousands, while sample size, i.e. the number of instances is much smaller. Such a setting affects the learning and predictive capabilities of machine learning algorithms and this phenomenon is known as the curse of dimensionality. Feature selection is a solution in such situations. This thesis is concerned with classification and feature selection in high dimensional data. It focuses on the design and application of methods achieving efficient feature selection which allows the estimation of models with good classific tion performance and stability of feature selection. This thesis proposes several means to handle the lack of enough samples in that high dimensional setting. Instance based learning, ensemble methods and prior knowledge based feature selection are the main concepts of our thesis contributions. Proposed selection methods have better classification performance and better stability.

Keywords: Feature selection, high dimensionality, stability, instance learning, ensemble methods, prior knowledge.