Disease Profiling of High–Dimensional Biomedical Data with Multiple Classifier Systems

by

Adenike Yewande Bamgbade

A thesis
Submitted to the Faculty of Graduate Studies
in partial fulfilment of the requirements for the
degree of

Master of Science

Department of Computer Science,
University of Manitoba Winnipeg, Manitoba

©Adenike Yewande Bamgbade 2005

THE UNIVERSITY OF MANITOBA

FACULTY OF GRADUATE STUDIES

*****

COPYRIGHT PERMISSION

Disease Profiling of High-Dimensional Biomedical Data with Multiple Classifier Systems

BY

Adenike Yewande Bamgbade

A Thesis/Practicum submitted to the Faculty of Graduate Studies of The University of Manitoba in partial fulfillment of the requirement of the degree

Of

Master of Science

Adenike Yewande Bamgbade © 2005

Permission has been granted to the Library of the University of Manitoba to lend or sell copies of this thesis/practicum, to the National Library of Canada to microfilm this thesis and to lend or sell copies of the film, and to University Microfilms Inc. to publish an abstract of this thesis/practicum.

This reproduction or copy of this thesis has been made available by authority of the copyright owner solely for the purpose of private study and research, and may only be reproduced and copied as permitted by copyright laws or with express written authorization from the copyright owner.

Abstract

The discovery of unique and biologically relevant biomarkers is important for the successful development of disease profiles from biomedical data. However, biomedical data are plagued by the curse of dimensionality and the curse of data sparsity because of their low sample to feature ratio. These dual curses are responsible for feature subset instability and overfitting during feature subset selection, thus making biomarker discovery difficult. A multiple-classifier-system approach to biomarker discovery is presented in this thesis to address the challenges of the dual curses to biomarker discovery. This approach incorporates an evidence accumulation based feature subset selection strategy into a random subspace method framework. Results from experiments on two real life spectral datasets show that the proposed solution strategy is able to obtain possible biomarkers.

Keywords Biomarker Discovery, Evidence Accumulation, Feature Extraction, Genetic Algorithms, External Crossvalidation, Random Subspace Method.

Acknowledgements

I would like to express my profound gratitude to my thesis supervisor, Dr. R. Baumgartner, for his guidance and suggestions in the course of this research. I am very grateful to Dr. Ray Somorjai for providing the opportunity to conduct my research at the Biomedical Informatics Unit of the Institute for Biodiagnostics, and for his invaluable suggestions as a member of my thesis examination committee. Also, I am grateful to Dr. Ben. Li for graciously agreeing to be on my thesis examination committee. I would also like to thank Drs Nikulin and Dolenko for their assistance during the implementation of the work. Special thanks go to Dr. E. Pranckeviciene for her suggestions and assistance, and especially the friendship and encouragement throughout my research. I am also grateful for the friendship and encouragement of Dr. Bryksina and Dr. Jeon.

I am grateful for the love and constant encouragement of my husband, Moses Owolabi, who never stopped believing in me. Thank you. I am grateful for the love, encouragement, and prayers of every member of our families (the Owolabis and the Bamgbades). I humbly acknowledge the support of numerous friends that space will not permit me to list by name. I am also grateful for the assistance provided by Dr. Ehikioya.

I gratefully acknowledge the financial assistance of the University of Manitoba in the form of the University of Manitoba Graduate Fellowship, and the research funding provided by the National Science and Engineering Research Council of Canada (NSERC).

Finally, I owe everything to the almighty God who saw me through it all. (Isaiah 50:7).

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Background and Problem Description

When a disease is present in an organism, its presence is evident on the molecular level in the form of changes in expressed gene levels, or the content level of protein and/or other biochemical substances in the biofluids and/or tissues of the organism. Novel non-invasive modalities such as microarray technology and various forms of spectroscopy generate biomedical data (magnetic resonance, infrared, fluorescence and Raman spectra from biotissues or fluids and mass spectra from proteomics), which give a quantitative assessment of the changes in gene expression levels, protein content levels, or the biochemical content and nature of biofluids and biotissues. Currently, there is significant interest in the application of pattern recognition algorithms to biomedical data to facilitate discrimination between diseases and disease states at the molecular level. This interest is stimulated by the fact that diseases are evident earlier on the molecular level than at the

morphological level. It is well known that the early identification or diagnosis of many deadly diseases can lead to early treatment, which will drastically improve prognosis. For example, Li and Ramamohanoar [32] note that only 35%–40% of advanced ovarian cancer patients have a five–year survival rate after treatment. On the other hand, patients whose conditions are detected early have a 95% five–year survival rate after treatment.

A contemporary direction in automated medical diagnosis is disease profiling. Disease profiling deals with the analysis of the distinguishing substances (gene expression levels, proteins, biochemical substances) in biofluids or biotissues of a specific disease, with the goal of characterizing the disease and generating a disease profile specific to the disease. An example of a disease profile generated from microarray data may be of the form; "if (gene g1 is over-expressed) AND (gene g37 is under-expressed) AND (gene 134 is very over-expressed) THEN most probably this is cancer type C (123 out of available 130 samples have this profile)" [29]. Obviously, an important requirement for disease profiling is the discovery of a subset of genes (in the example genes g1, g37, and g134), proteins or biochemical substances typical to all or a majority of the biofluids and biotissue samples of the specific disease. More importantly, the subset of substances must be biologically interpretable to be useful to researchers. That is, a causal relationship between the subset of substances and the disease should be obvious. Also, it will facilitate drug development and studies on patients' therapeutic responsiveness to drugs. These subsets of substances are referred to as biochemical markers or biomarkers [50]. The identification and validation of potential biomarkers for ratification and inclusion into disease profiles is referred to as biomarker discovery.

Srinivas et al. [51] define biomarkers as

> *"biological molecules that are indicators of the physiologic state and also change during a disease process."*

Biomarkers are useful for identifying and detecting a disease at its early stage; for monitoring the progression of the disease; and serve as a disease indicator [51]. Biomarker discovery involves the identification and validation of substances that exhibit causal links with biofluid or biotissue samples of a particular disease. These substances may be identified due to a marked difference between their content levels in the diseased tissues and their content levels in healthy tissues. In addition, these substances may be distinguishable as a result of modifications in their chemical structure.

Biomarker discovery is akin to feature selection or feature extraction in biomedical data with pattern recognition algorithms. The goal of biomarker discovery is to select a subset of genes, proteins or biochemical substances that can be used to correctly determine the class of previously unclassified biotissues or biofluids. In biomedical datasets, the biofluids and biotissues are represented as a sequence of values, such as the gene expression levels, protein levels, or mass-to-charge ratios of the biosubstances.

Biomedical data are characterized by a large number of features (in the order of thousands and tens of thousands) and comparatively fewer samples (usually a few dozens or less). In other words, biomedical datasets are high-dimensional and sparse. An exhaustive search through all possible biomarkers for an optimum biomarker in such high-dimensional datasets is computationally intractable. Consequently, feature subset selection algorithms that search for near-optimal biomarkers are used for biomarker discovery

in biomedical datasets. Unfortunately, the large discrepancy between biomedical data dimensionality and sample size makes it susceptible to the curse of dimensionality [6]. The curse of dimensionality is responsible for overfitting [30] – the discovery of poor quality feature subsets in high-dimensional data. It can be eliminated by ensuring that the dataset size satisfies the sample to feature ratio (SFR) requirement of 5 or higher [28]. However, achieving such SFR costs lots of money and time, and may sometimes be outright impossible.

In addition to the curse of dimensionality, the sparseness of biomedical datasets renders them victims of the curse of sparsity. The curse of sparsity is responsible for solution instability in feature subset selection with respect to perturbations in the data [48, 49]. Solution instability in feature selection, also known as feature subset instability, refers to selection of the different sets of features from different partitions of the data, with the same feature selection parameters. Solution instability cannot be solved by complying with the SFR requirement.

An important requirement for an ideal biomarker is that it must be specific [7]. A biomarker is specific when it identifies a particular disease and is reproducible. Consequently, feature subset instability and overfitting pose a difficult challenge to the discovery of a potential biologically relevant set of biomarkers for disease profiling from high-dimensional and sparse biomedical data.

1.2 Research Motivation and Objectives

Solution instability and overfitting are problems that occur also in predictive classification. Classifiers learned on small-sized datasets are known to be unstable and to overfit. Multiple-classifier systems (MCSs) are used as a stabilizing technique for unstable classifiers and to eliminate the effect of overfitting. A multiple classifier system consists of a set of individually trained classifiers, whose individual decisions are combined to determine the class of new samples. Research shows that combining the decision of unstable classifiers results in solution stability and eliminates overfitting. MCSs are based on the concept of evidence accumulation [17, 56]. This concept seeks to combine the "opinions" of multiple "experts" (classifiers, in the case of MCS) to achieve the best possible result.

In this thesis, I explore the applicability of evidence accumulation to feature selection for biomarker discovery, and the discovery of the important biofeatures that contribute to the stability and accuracy of a MCS. This research is motivated by the notion that success with evidence accumulation in other domains for eliminating solution instability should be possible in feature selection. Also, with the success of MCSs, I expect that the features that contribute to the stability and classification accuracy of a MCS will occur more often in the feature spaces of its constituent classifiers. Consequently, I expect that the co-occurrence of features that appear in optimal feature subset selections used to create an accurate multiple classifier system, should lead to the discovery of a set of highly discriminatory and biologically relevant biomarkers for further investigation as disease profiles. I would like to emphasize that the discovery of the important features

that influence the performance of an MCS has been identified as an important unsolved issue [12].

Therefore, the objectives of this research are as follows:

1. To develop an evidence accumulation framework for feature selection,
2. To use the evidence accumulation framework for biomarker discovery in high-dimensional biomedical datasets,
3. To develop a strategy for discovering the features that contribute to the accuracy of a MCS, and
4. To use the strategy to extract potential biomarkers from high-dimensional biomedical datasets.

It is important to note that validating the biological interpretability of the discovered potential biomarkers is only possible with the expertise of a domain specialist. Therefore, the potentially important findings in this research require further validation by domain experts to establish the clinical applicability of these features. My goal is to discover potential biomarkers with this proposed solution strategy. Validating the biological relevance of the discovered biomarkers is beyond the scope of this research.

1.3 Thesis Layout

The layout of this thesis is as follows: Chapter 2 reviews previous work related to the proposed research. The evidence accumulation framework for feature selection and its

application to biomarker discovery are presented in Chapter 3. In Chapter 4, random subspace method feature selection (the MCS feature discovery strategy) and its application to biomarker discovery are presented. The conclusions and recommendations are presented in Chapter 5.

Chapter 2

Related Work

2.1 Introduction

In this Chapter, I review literature relevant to the research in this thesis. A review of feature subset selection and its application to biomarker discovery is presented in Section 2.2. The challenges of feature subset selection for biomarker discovery are highlighted. Section 2.3 focuses on research work related to biomarker discovery using wrapper-based feature subset selection algorithms; the shortcomings of these attempts were highlighted and discussed. Section 2.4 presents a discussion on cross validation - the evaluation technique used in this thesis. Evidence accumulation as a solution to feature subset instability is discussed in section 2.5. The Chapter concludes with a summary.

2.2 Feature Subset Selection for Biomarker Discovery in Biomedical Data

An important prerequisite for the development of disease profiles is the discovery of a subset of substances in biofluids or biotissues that show causal links with the disease present in the biofluids or biotissues. Currently, feature subset selection algorithms applied to biomedical data have been used in many studies for biomarker discovery in biomedical data, especially in the area of gene selection from microarray gene expression levels for cancer profiling [2, 5, 14, 21, 24, 52, 55, 57, 59], and spectral region selection in datasets of mass spectra, infrared spectra, and magnetic resonance spectra [48].

From a supervised pattern recognition standpoint, the biofluid or biotissues are feature vectors, whereas the gene levels, mass-over-charge ratios, or spectral intensities are the "explanatory" features of the feature vectors. Given a biomedical dataset L consisting of N biofluid or biotissue samples of the form $(x_1, y_1), ..., (x_N, y_N)$ where x_i is a d-dimensional vector of features of sample i, and y_i is the output label of sample i signifying the class or type of the sample (e.g. diseased or healthy), the goal of the feature subset selection algorithm is to select a subset m of features from the d-dimensional feature vector x_i (where $m << d$) such that the subset m can predict the output y_j of a previously unlabelled sample $(x_j, *)$ with equal or higher accuracy than when the entire d-dimensional dataset is used.

There are two approaches to feature subset selection: filter methods and wrapper methods [30]. Both approaches are currently used for biomarker discovery [53]. The

main difference between these methods is that while wrapper methods evaluate features subsets on the algorithm that will be used to build the final classifier, filter methods do not.

Filter methods select and evaluate feature subsets on the basis of the individual feature's value with respect to the output labels of the data using statistical methods such as T-tests and F-tests. A common approach adopted by filter methods is to rank the features in decreasing order of their individual value and then select the top ranking set of features as the selected feature subset. Filter methods are preferred over wrapper methods for gene selection because they are computationally more efficient [52]. However, Guyon and Elisseeff [23] note that feature ranking may lead to the discovery of redundant subsets of features. Also, they point out that feature ranking may lead to the elimination of a feature that may appear useless when ranked individually, but may be important when considered in a subset along with similarly "unimportant" features.

Wrapper methods search for possible subsets of features in the entire feature space. The search is guided by the accuracy of the selected subset on a learning algorithm. Crossvalidation is often used to evaluate the fitness of the feature subsets in small datasets, because of the limited amount of data available. Some important issues a wrapper method needs to address include: how to search the feature space; a learning algorithm for assessing the feature subsets and guiding the search; and a criterion for stopping the feature search. A brief review of the feature subset search techniques is presented in section 2.2.1.

Although wrapper methods tend to be more computationally intensive than filter

methods [52], Kohavi and John [30] show that wrapper methods produce feature subsets with better predictive accuracy than filter methods. Xiong et al. [58] show that wrapper-based methods are better than filter methods for biomarker identification.

2.2.1 Feature Subset Search Techniques

Wrapper-based feature selection techniques do not rank features on the basis of each feature's individual importance. Rather, greedy or heuristic search techniques are used to select feature subsets, which are evaluated for their importance, usually on the basis of their misclassification error with a classifier (see 2.2.2). Two techniques that guarantee the discovery of the optimal feature set are: an exhaustive search or the branch and bound technique. An exhaustive search examines all possible feature subsets in the feature space in its search for the optimal feature subset. As a result, exhaustive searches are only possible with very small feature spaces. For biomedical datasets, which have large feature spaces (thousands of features), an exhaustive search is computationally intractable.

The branch and bound algorithm guarantees the optimum feature subset without an exhaustive search. This technique is based on the assumption that the feature subset selection criterion satisfies the monotonicity property. Given a set of nested feature subsets $X_1, X_2, X_3, ...X_j$, and a criterion function $C(.)$, if $X_1 \subset X_2 \subset X_3... \subset X_j$, then the monotonicity property requires that the performance of the criterion function should improve with the addition of more features, that is $C(X_1) \leq C(X_2) \leq C(X_3)... \leq C(X_J)$. Jain and Zongker [27] note that the branch and bound technique is unsuitable for datasets

with large feature to sample ratios (as is the case in biomedical datasets). They note that the curse of dimensionality (exhibited by high-dimensional biomedical datasets) violates the monotonicity condition. Consequently, the branch and bound technique is not suitable for wrapper based feature selection in biomedical datasets.

Other commonly used search strategies, such as the sequential forward selection (SFS) and sequential backward selection (SBS) arrive at near-optimal solutions. The sequential forward selection (SFS) starts its subset search from a single feature subset and progressively adds features to the feature subset until the search termination criterion is satisfied. The sequential backward selection (SBS) starts with the full set of features and systematically eliminates unnecessary features until the selection termination criterion is satisfied. A major drawback of SFS and SBS is that they do not backtrack. That is, once a feature is selected by a SFS it may not be discarded later. Also, once a feature is discarded in an SBS it may no longer be considered for inclusion later. Extensions to SFS and SBS that allow backtracking include sequential forward floating search, sequential backward floating search, and plus-one-take-way-r.

Biologically inspired processes such as genetic algorithms (GAs) are also used for feature subset search. GA usually starts its feature subset search from a set of possible solutions, which it modifies to achieve the best possible feature subset. The solution arrived at by the GA is referred to as a chromosome. The GA starts by generating a possible set of solutions (chromosomes), which are referred to as the *population*. Through operations similar to crossover and mutation, the population evolves into progenies with a better chance of survival. The progenies evolve through a number of user-defined cy-

cles known as *generations*. A crossover produces an offspring from a synthesis of parts of two parent chromosomes. Mutation alters parts of a chromosome to arrive at an almost identical chromosome. The goal of the GA is to evolve chromosomes that will optimize the feature subset selection criterion. At each cycle the evolved progenies are evaluated using the selection criterion. Siedlecki and Sklanksy [40] note that the mutation operation is instrumental in preventing GA from converging early to poor local minima.

2.2.2 Classifiers

The error rate of a classifier is the most commonly used feature subset selection criterion. In this thesis, I am interested in supervised classifiers. Given a set of e training samples of the form $(x_1, y_1), ..., (x_e, y_e)$ where x_i is a d-dimensional vector of features of sample i, and y_i is the output label of sample i signifying the class or type of the sample (e.g. diseased or healthy), a classifier is a function $y = f(x)$ that correctly assigns the value y_* to a previously unknown sample x_* from the same underlying data distribution as the e training samples. Jain et al. [26] identify the following three approaches to classifier design:

1. Classifier design based on the concept of similarity - These classifiers assign samples to classes on the basis of their pattern similarity. A similarity metric for comparing samples is required, and an appropriate choice is crucial for the success of the classifier. The nearest neighbour classifier is an example of a classifier based on this design concept,

2. Classifier design based on the probabilistic approach - These classifiers usually have the classification problem posed in probabilistic terms, and the probability values are known. A typical example of a classifier based on this design concept is the optimal Bayes decision rule, which assigns classes to samples on the basis of their posterior probability, and

3. Classifier design based on the construction of decision boundaries by optimizing a certain error criterion such as the mean squared error between the classifier output and the true label of the sample. Linear discriminant analysis and support vector machines are two widely used examples of classifiers in this group that are used in this research, and will be discussed subsequently.

Linear Discriminant Analysis

The discussion on linear discriminant analysis (LDA) presented in this section is based on the works of Jain et al. [26] and Duda et al [15]. In the two-class case, the goal of LDA is to find the decision boundary of the form $g(x) = w_i^t x + \omega_o$ that best partitions the sample space into two regions; where w^t is the weight vector and ω_o is the threshold weight. When $g(x)$ is linear in x, it is a linear discriminant function, which serves as a classification rule, such that given a new sample x^*, it assigns x^* to class 1 if $g(x) > 0$ and to class 2 otherwise. Linear discriminant analysis is based on the assumption that the samples to be partitioned have a gaussian distribution. Given a dataset with samples x_i from one of 2 classes C_1, C_2, LDA attempts to find the weight vector w^t such that the projected clusters of C_1 and C_2 samples on w^t are well separated from each other, while

the variance between the samples in each class cluster is small. This can be achieved by maximizing the following objective function

$$J(w) = \frac{w^t S_B w}{w^t S_W w}$$

with respect to w^t where S_B is the between-class scatter matrix, and S_W is the within-class scatter matrix defined as

$$\begin{aligned} S_W &= \sum_{x \in C_1} (x - m_1)(x - m_1)^T + \sum_{x \in C_2} (x - m_2)(x - m_2)^T \\ S_B &= (m_1 - m_2)(m_1 - m_2)^T \end{aligned}$$

where m_i is the sample mean for class i samples. The optimal W is taken as the eigenvector of the largest eigenvalue resulting from the eigenanalysis of the relation $S_W^{-1} S_B$. LDA works with the notion that the underlying distribution of the data is Gaussian, and the data is separable. However, they are known to perform well with datasets that do not have a Gaussian distribution.

Support Vector Machines

The discussion on support vector machines presented in this section is based on the work of Jain et al. [26]. The Support vector machine (SVM), also known as support vector classifier, is a new classifier that has been shown to work well on separable and nonseparable data. SVM is essentially a two-class classifier that seeks to optimize the distance between the decision boundary that separates the two classes, and the training samples that are closest to the decision boundary. These closest training samples are the support vectors, and the distance between the decision boundary and the support vectors is called the margin (see Figure 2.1).

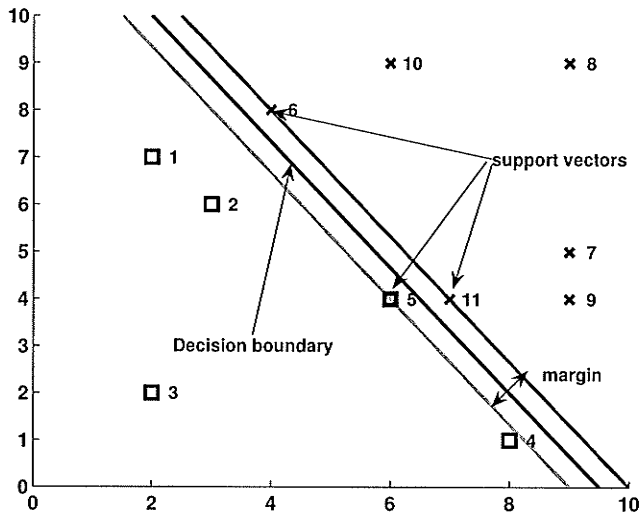

Figure 2.1: An example of an SVM decision boundary for a two-class problem. The squares are class 1 samples while the crosses are class 2 samples

Maximizing the margin minimizes the number of support vectors in the decision function. The decision function derived by SVM for a two-class problem may be of the form:

$$D(x) = \sum_{\forall x_i \in S} \alpha_i \lambda_i K(x_i, x) + \alpha_0,$$

where S is the set of support vectors from the training set, $\lambda_i = \pm 1$ the labels of the feature vectors x_i, and $K(x_i, x)$ is the kernel function. The values of $\alpha_i \geq 0$ are obtained from the training set by optimizing

$$\min(\alpha^T \Lambda K \Lambda \alpha + C \sum_j \xi_j)$$

$$subject\ to\ \lambda_i D(x_j) \geq 1 - \xi_i, \forall x_j$$

Λ is a diagonal matrix containing the labels λ_j and the matrix K stores the values of the kernel function $K(x_i, x)$. ξ is the set of slack variables that allows class overlap in the

margin. The degree of overlap is controlled by the value $C > 0$. During optimization only the values of the support vectors are non-zero; all other values of α_i become 0. Four commonly used kernel functions $K(x_i, x)$ that can be found in literature are the linear function, the polynomial function, the radial basis function, and the sigmoid function.

2.3 Wrapper-based Feature Subset Selection from Biomedical Data for Biomarker Discovery

Wrapper-based feature selection has been used for biomarker discovery in biomedical data in a number of studies. Xiong et al. [58] use two feature subset selection algorithms, sequential forward search (SFS) and sequential forward floating search (SFFS), to search for potential gene biomarkers in three publicly available gene microarray datasets. Their choice of biomarkers was dependent on the predictive accuracy of the gene subsets on the following classifiers - LDA, SVM, and Logical regression classifier (LR) using the leave-one-out crossvalidation procedure. Their results show that a similar set of features displayed high predictive accuracy on the three learning algorithms, although the degree of accuracies varied from one learning algorithm to the other. In [59], four feature search strategies - sequential forward selection, Monte Carlo methods, T-tests, and Golub's prediction strength statistic [21] were used to search for discriminatory genes from microarray datasets. The classification accuracy of the selected gene subsets on a Fisher linear discriminant function was used to assess and compare the selected genes. Xiong et al. [59] note that their forward stepwise procedure and Monte Carlo methods

selected feature subsets that performed better than features selected from the T-test and prediction strength feature rankings. They showed that feature subsets selected with the Monte Carlo methods had classification accuracies as high as 93% . Xiong et al. [59] note that the gene subsets selected using the forward stepwise procedure and Monte Carlo methods are not unique but have classification accuracies that are quite close.

Iñza et al. [52] use SFS for selecting features from a microarray dataset. They assess the features selected by the SFS on the following classifiers, an IB1, a naive Bayes classifier, C4.5, and CN2. Although their focus is not the discovery of biologically significant genes, their results show that an accurate gene subset, which may be biologically significant, can be found in microarray datasets using wrapper based feature selection approaches.

Li et al. [33] use genetic algorithms (GA) and K-nearest neighbour (KNN) to select a subset of predictive genes from a microarray dataset. They use the GA to search for possible gene subsets while the fitness of the gene subsets is measured by their predictive accuracy on the KNN algorithm using crossvalidation. Their results show that they discover a predictive subset of gene on repeated trials on the same dataset. However, they show that on different partitions of the dataset, the subset of features selected differed significantly. They attribute this observation to the small sample size of the datasets used in the study.

Wrapper based feature selection algorithms have been used for biomarker identification in spectroscopy datasets. Baggerly et al. [3] use GA to search for feature subsets and assess the selection using an Euclidean distance measure. Levner [31] also uses SFS, a

modified version of SBS, and a version of boosting (boostedFE), extended for feature selection, to select discriminatory features and possible biomarkers from five proteomic mass spectroscopy data. The objective of his study was to evaluate the suitability of the above mentioned techniques, and some filter feature selection techniques, for feature selection for spectra data. His results show that SFS and boostedFE outperformed all the other techniques. He showed that only boostedFE was consistent on all five tested datasets used in his study.

Although, the work reviewed so far shows that wrapper based feature selection algorithms can be used to select potential biomarkers with high classification accuracy from biomedical datasets, the biological significance of these selected feature subsets is questionable because of their lack of uniqueness on different dataset partitions. Li et al. [33] show that significantly different gene subsets were selected from different partitions of a gene microarray dataset. However, they note that on the same dataset partition repeated feature searches selected the same gene subset. Similarly, Somorjai et al. [48] show that multiple "optimal" feature subsets with equally high or identical prediction accuracies can be selected from different partitions of a spectral dataset. Li et al. [33] and Somorjai et al. [48] attribute this observation to the small sample size and sparsity of the biomedical datasets. Simon and Altman [45] also show that most of the results from different biomarker identification studies are often inconsistent or contradictory.

For wrapper based feature selection to be successfully used for biomarker discovery, the selected feature subsets should be stable irrespective of dataset partitions. Given the constraint of the small and restricted size of biomedical datasets, there is a need to

develop a feature subset selection strategy that will eliminate the feature subset instability and overfitting in wrapper based feature selection from high-dimensional small and sparse biomedical datasets.

2.4 Crossvalidation

Crossvalidation (CV) is a technique for estimating the generalization error associated with a model building algorithm [41]. Crossvalidation is especially useful for biomedical datasets, which are difficult to partition into reasonably-sized portions as separate training sets and test sets for model fitting. Crossvalidation involves the splitting of data into k equal parts. One of the partitions is set aside as the test set while the other $k-1$ parts together form the training set. The training set is used to fit the model, and the test set is used to estimate the accuracy of the fitted model. This process is repeated for each of the k parts and the accuracy of the model is taken as the average performance of the accuracy over all the k test sets. The case $k = N$, where N is the total number of samples in the dataset, is known as leave-one-out crossvalidation. Common choices of the value of k for crossvalidation are 5 and 10.

Ambroise and McLachlan [1] and Simon [44] show that previous applications of crossvalidation in studies on biomedical datasets preceeded the application of CV by feature subset selection. The entire dataset was used to select a discriminatory feature subset that was used to develop the CV model estimates. The developed models were subsequently evaluated using CV. Ambroise and McLachlan [1] and Simon [42] refer to this approach to CV as internal crossvalidation and show that it is not a proper prediction

error estimation technique because it is open to selection bias. They show that excluding the feature subset selection from each step in the CV results in an optimistic estimate of the generalization error of the model-building algorithm. They note that proper crossvalidation must include feature selection in each step of the process. They refer to this CV strategy as external crossvalidation and show that it results in a more unbiased estimate of the generalization error by reducing selection bias.

Simon [41] notes that CV provides an unbiased estimate of the generalization error of a model building algorithm and not that of the resulting model, as widely held by computer scientists and statisticians. He states that small, properly crossvalidated error rates are significant [42]. However, he notes that further clinical validation is required to confirm these results for clinical applicability. Simon [43] also states that proper CV error estimation can be used to determine if a model building algorithm overfits. He notes that algorithms that overfit will not have low crossvalidated error estimates if crossvalidation is performed properly [43].

2.5 Evidence Accumulation

The solution strategy presented in this thesis is based on the concept of evidence accumulation (EA). The application of EA in this research is inspired by the work of Fred and Jain [18] on evidence accumulation clustering for combining multiple clusterings to arrive at a consistent data partition. According to Fred and Jain [18]

"The idea of evidence accumulation clustering is to combine the results of

multiple clusterings into a single data partition by viewing each clustering result as an independent evidence of data organization."

Evidence accumulation seeks to combine independent evidence - multiple solutions to a problem, to arrive at a single solution (possibly the optimum solution to the problem).

Fred and Jain [18] note that the following three requirements must be satisfied for evidence accumulation:

- How to generate/collect evidence,
- How to combine the evidence, and
- How to extract a single final solution from the combined evidence.

An important success factor for evidence accumulation application is that the opinions to be combined must be diverse and accurate in their own rights. Fred and Jain [17] present possible ways of generating the required diverse opinions for evidence accumulation in data clustering:

- by combining the results of different (clustering) algorithms,
- by producing and combining different results generated by a single algorithm on different partitions of a dataset, and
- by running an algorithm on the same data, but with different initialization parameters.

Sensor fusion [39] and multiple classifier systems [13] are other successful applications of the evidence-accumulation concept. Evidence accumulation in the form of multiple classifier systems has been known to mitigate overfitting and solution instability in predictive classification. These problems are the outcome of the curses of dimensionality and sparsity, which have been identified as the culprits of feature subset instability and overfitting in feature selection. Multiple classifier systems are discussed in the next section.

2.5.1 Multiple Classifier Systems

A multiple classifier system (MCS) consists of a set of individually trained classifiers, whose individual decisions are combined to determine the class of new samples. As mentioned earlier, MCSs are based on the concept of evidence accumulation [17, 56]. MCSs have received significant attention in the literature because of their ability to reduce the variability and improve the generalizability of classification. A good MCS should consist of classifiers that differ in their predictions [36]. Dietterich [13] notes that the accuracy of a good MCS is usually higher than the accuracy of any of its constituent classifiers. MCSs have also been observed to serve as a stabilizing technique for weak classifiers that show high error variability [46].

Typically, a multiple classifier system consists of two or more classifiers and a combiner or combination technique that merges the decisions of the ensemble into a final decision. The ensemble may consist of classifiers trained with different learning algorithms, or all the classifiers may be trained with a single learning algorithm. For MCSs

based on a single learning algorithm, the required diversity between the constituent classifiers is created by making the data used to train the classifiers diverse. Classifier diversity may be created by manipulating or subsampling the training data, by manipulating or subsampling the features of the training data, by manipulating the output labels of the data, or by infusing randomness into the learning algorithm used to train the classifiers [13]. Examples of MCS combining methods include unweighted voting, weighted voting, and gating networks. An unweighted vote takes the decision of the majority in the classifier ensemble as the decision of the ensemble. A weighted vote qualifies the vote of each classifier in proportion to the accuracy of the classifier. Consequently, the decision of the ensemble is determined by both the decision of the majority in the ensemble and the weights of their decisions. A gating network computes weights for each classifier in the ensemble using a feature vector, then uses these weights in a weighted vote for the ensemble.

Another approach to MCS creation presented by Somorjai et al. [47] combines the weights of an ensemble of classifiers into the input for a single final classifier. Their approach makes N random training set-test set splits of the data. The training set is used to learn a classifier and the test portion is used to evaluate the classifier. The weighted average of the coefficients of the N classifiers is used to create the final classifier. The weight assigned to each classifier is a product of its Cohen's chance-corrected measure of agreement and a value which may be equal to either 0 or 1 depending on the proportion of the classifier's crisp assignments.

Bagging (bootstrap aggregating) [8], boosting [19], and the random subspace method

(RSM) [25] are some of the popular methods for creating MCS. Bagging works by aggregating, with a majority vote, the decisions of classifiers trained on bootstrap samples [16] of the data. A bootstrap sample of the data is generated by sampling, with replacement, N instances of the training set (where N is the total number of instances in the training set). Consequently, each bootstrap sample may contain multiples of a particular instance of the training data, or no representation of some instances. Boosting is similar to bagging in that it votes over the decisions of its constituent classifiers. However, its classifiers are generated sequentially, by adjusting the weights of instances of the training sets of subsequent classifiers based on the performance of preceding classifiers. RSM generates a classifier ensemble from the training set by randomly selecting a subset of the features in the training set. As is the case with bagging and boosting, RSM votes over the decisions of the classifiers to arrive at a final decision.

In [46], Skurichina evaluates the effectiveness of bagging, boosting, and RSM as MCSs (with linear classifiers as base classifiers) and their stabilizing effect. Although linear classifiers (LDA) have been deemed unsuitable as base classifiers for these MCS techniques because of their stability, Skurichina [46] shows that linear classifiers trained on small sample sizes are unstable and hence suitable. Her conclusions show that all three methods are effective as multiple classifier systems with linear classifiers as their base classifiers. However, she concludes that only RSM is a stabilizing technique.

2.5.2 Feature Selection for Multiple Classifier Systems

The success of the RSM has inspired the use of feature subset selection for the creation of MCSs. Gunter and Bunke [22] present a feature selection strategy for MCS creation. Instead of randomly selecting feature subsets for the ensemble creation, as is the case with RSM, they use a feature selection algorithm to select well-performing feature subsets for each classifier in the ensemble. After each subset selection, the selected subset is included in a list of "forbidden feature subsets". The features in the forbidden list are not eligible for selection in the subsequent rounds of feature subset selection. Their strategy ensures the creation of a highly accurate and diverse set of classifiers required for a successful MCS. The feature selection algorithm used in their experiments is a modified version of the "plus–1–take–away–one" algorithm [38]. However, they note that any feature selection algorithm will suffice. Their results show that their ensemble creation method performed better than RSM, adaboost, and bagging. Opitz [36], Olivera et al. [35], and Tsymbal et al. [54] have also used feature selection for MCS creation with results that surpass those of traditional MCS creation techniques.

As far as I know, evidence accumulation has not been applied to supervised feature selection. It is possible that if evidence accumulation can mitigate overfitting and solution instability in predictive classification, it may serve as a solution to feature subset instability and overfitting in feature subset selection. Consequently, it will facilitate the discovery of reliable biomarkers in high-dimensional biomedical data.

From the discussion so far, it is clear that MCSs based on well performing feature subsets perform better than traditional MCSs that do not involve feature subset selection.

Consequently, the stability and accuracy of MCSs may be attributed to the strength of the important features in the feature subspaces of its constituent classifiers. I expect that the features that contribute to the stability and classification accuracy of the MCS will occur more often in the feature subspaces of its constituent classifiers. Consequently, it is expected that the co-occurrence of features that appear in optimal feature subset selections, used to create an accurate multiple classifier system, should lead to the discovery of a set of highly discriminatory, stable, and biologically interpretable biomarkers. However, Dietterich [12] notes that MCS does not provide insight (that is, the features responsible) into how it achieves the desired accuracy and stability. In single classifiers, the user can easily identify the features that influence the decisions of the classifiers. These features are obvious either as a result of the weight the fitting algorithm assigns to these features, or as a result of the inclusion of the features in the classifier. However, in a multiple classifier system, where different weights may be assigned to a particular feature in different classifiers, or the features included in the ensemble vary, it is difficult to determine the influence of each feature and the magnitude of this influence on the MCS.

2.6 Summary

In this chapter, I presented a review of feature subset selection and its application to biomarker discovery. Wrapper-based feature subset selection algorithms were shown to be the better choice for biomarker discovery. Solution instability and overfitting were highlighted as the two important drawbacks to the application of wrapper based feature subset selection for biomarker discovery in biomedical data. From the literature

reviewed it is clear that evidence accumulation has been successfully applied to solve solution instability and overfitting in predictive classification. Therefore, by adopting evidence accumulation in feature subset selection, I hope to discover stable and robust feature subsets that may be possible biomarkers. The next chapter discusses the evidence accumulation based feature selection strategy.

Chapter 3

Evidence Accumulation-Based Feature Selection

3.1 Introduction

In this Chapter an evidence accumulation based feature selection strategy is presented. The details of this strategy are also published in [4]. My goal is to combine the results of multiple feature subset selections to arrive at a single result. The feature subsets are the independent evidence of discriminatory features in the dataset. A feature frequency histogram is used to combine the evidence, and a thresholding technique is used to extract a single final feature subset from the histogram. The biomedical datasets used in this research are discussed in section 3.2. The GA-ORS [34] is the feature extraction algorithm used in this research for evidence generation. In section 3.3 a discussion on the GA-ORS is presented and the reason for its use in this research is explained. Some

experiments were carried out to investigate the characteristics of the GA-ORS, and to determine some important parameters of the algorithm required for its role in the strategy. These experiments and their subsequent results are presented in section 3.4. The chapter concludes with a discussion of the results of the evidence accumulation feature selection strategy.

3.2 Datasets

Two MR-spectral datasets were used in the experiments discussed in Chapters 3 and 4. They are referred to as Dataset 1 and Dataset 2 respectively. The properties of the datasets and their original partitions into training and independent test sets are presented in Table 3.1. Both datasets contain spectral information on two classes of pathogenic fungi samples referred to as class 1 and class 2 respectively. Each of the samples are described by a 1500-feature vector. Each original feature is a spectral intensity.

Table 3.1: Properties of the Datasets used in this research

	Dimensionality	**Training Samples (Class 1) + (Class 2)**	**Test Samples (Class 1) + (Class 2)**
Dataset 1	1500	(62) + (62)	(42) + (31)
Dataset 2	1500	(70) + (70)	(105) + (59)

3.3 GA-Guided Optimal Region Selection (GA-ORS)

The feature selection algorithm used in the proposed solution strategies and subsequent experiments in Chapters 3 and 4 is the genetic-algorithm guided optimal region selector (GA-ORS) [34]. The GA-ORS is a wrapper-based feature extraction algorithm developed at the Institute for Biodiagnostics. GA-ORS is the algorithm of choice in the experiments discussed below, because unlike other feature extraction algorithms, it includes a preprocessing step that maintains the original features, but reduces the feature space size during subset evaluation. Consequently, the features returned by the GA-ORS are not transformed forms of the original features in the dataset. In addition, because it is a GA-driven algorithm, it is flexible and enables the feature subset search to move far from the location where the search started in the high-dimensional feature space. As a result, GA-ORS tends to avoid getting stuck in local minima. Another important feature of the GA-ORS is that it selects, via averaging adjacent features, subregions of features not individual features.

In the GA-ORS the fitness function to be optimized for guiding the search for discriminatory spectral regions is the mean square error of the training set classification, using linear discriminant analysis (LDA) with internal leave-one-out (LOO) crossvalidation. The input parameters of the GA-ORS are the maximum number of desired regions/feature subsets, the number of generations, the size of the initial population, the probability of mutation, the probability of crossover, and a random seed value for generating the initial population.

The operations of the GA-ORS is as follows. To select M regions, with the number

of generations set to value G and the population size set to the value P, the algorithm starts by generating P binary strings of length K, (where K is the number of features in each feature vector in the training set). M subregions of each of the P strings are randomly initialized to contiguous values of 1 while the other locations are initialized to zeros. The contiguous values of 1 represent a subregion in the feature space of the dataset. In each of the P strings, the M subregions are processed into M features and evaluated on the fitness function. The processing may entail taking the average over the attribute values in a subregion such that the subregion is represented by a single feature (that is the average). The P fitness values are sorted in ascending order, and the top E are marked as the elite. The elite E strings are automatically included in the next generation, while the non-elite $P - E$ strings are bred through mutation and crossover to derive the next generation. The algorithm terminates when the number of generations equal G.

3.3.1 Experimental Investigations of the Properties of the GA-ORS

The important characteristics of the GA-ORS were investigated experimentally to facilitate an understanding of its performance under certain conditions, and when some of its parameters are varied. The findings from this investigation will influence my choice of parameters for the GA-ORS in its role as evidence generator. I am interested in:

- The number of generations at which the GA-ORS starts to overfit,
- the stability of the feature subsets selected by the GA-ORS with different initializations and/or the same initialization, and

- the stability of the feature selection with different partitions of the dataset.

Two (2) experiments (experiment 1 and experiment 2) were conducted in this investigation to investigate the important properties of the GA-ORS essential for its role in the proposed methodologies. These experiments are discussed in the subsequent subsections.

These issues are important for the successful evaluation of the solution methodologies discussed subsequently.

Experiment 1

The objective of this experiment is to investigate the number of generations at which the GA-ORS starts to overfit the data. Overfitting occurs when a learning algorithm starts to fit a model to the peculiarities of a model especially when the learning procedure takes too long or the amount of available data is limited. A simple way of determining the onset of overfitting in a learning algorithm is to divide the data into a training set and a validation or test set. The learning algorithm is used to fit a model on the training set and the performance of the fitted model is evaluated on the test set. With each "training-and-evaluation" step, the complexity parameter of the learning algorithm is incremented. The learning algorithms starts to overfit the model when the training error decreases while the validation error increases. Figure 3.1 (used from [20]) illustrates the learning curves of a hypothetical learning algorithm, which has time as its complexity parameter. From the figure, we observe that overfitting starts to occur when the learning algorithm's execution exceeds t. Hence, t is a possible optimum value for the algorithm to avoid overfitting.

The complexity parameter I am interested in is the optimum number of generations that the GA-ORS must iterate through before it starts to overfit.

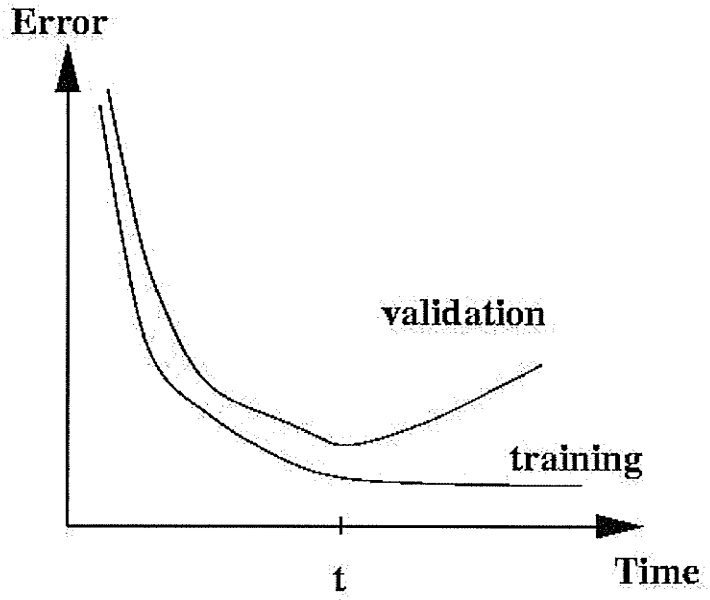

Figure 3.1: Measuring overfitting[20]

This experiment investigates the stage at which the GA-ORS starts overfitting as the number of generations increase. This experiment is carried out on both Dataset 1 and Dataset 2. The experimental procedure is as follows. Each dataset is randomly split into 25 training and test set partitions in accordance with the configuration presented in Table 3.1. The GA-ORS is used to select three (3) feature regions from each training set, and the probabilities of crossover and mutation are set at 0.6600 and 0.00100 respectively (these values are used in all the experiments in Chapters 3 and 4). The GA-ORS is initialized with a constant seed value while the number of generations of iteration is increased from 10 to 100 in steps of 10. The feature subsets selected from each of the 25 training set splits are used to learn LDA classifiers, which are then used to classify

their corresponding test set samples. The misclassification errors of the 25 splits at each generation was recorded. The external crossvalidation (ECV) error of over the 25 splits was also estimated by taking the mean error of the 25 splits and the standard deviation. Also, the 3-fold crossvalidation error of each of the 25 split training sets were carried out 10 times. Each of the ten trials was based on random splits of the training sample into 3 folds. The ECV recorded for each split was taken as the mean over the 10 trials.

Experiment 2

This experiment has two objectives: to investigate (1) the effect of varying the initialization of the GA-ORS on the stability of the selected feature subsets and (2) the stability of the feature subsets selected by the GA-ORS from different partitions of a dataset. Five (5) out of the 25 dataset splits of Dataset 1 and Dataset 2 used in experiment 1 are used in this experiment. In this experiment, 25 different selections of 3 regions were made via the GA-ORS from each of the 5 dataset splits. The GA-ORS was initialized with different randomly selected seed values for each of the 25 selections. The occurrence frequency of the features selected in 25 selections was recorded and a histogram of these frequencies plotted. Again, 25 feature selections were made from each of the 5 splits of the datasets, however, a constant seed value was used to initialize the 25 selections.

3.3.2 Results of GA-ORS Investigations

Figures 3.2 and 3.3 illustrate graphs of the test errors and ECV errors on both datasets from experiment 1. The graphs show that the ECV error estimate is more pessimistic

than the test error. The results of the investigations on the GA-ORS show that the external crossvalidation error of feature subset selections from both Dataset 1 and Dataset 2 improves (that is, decreases) as the number of generations increase from 10 to 30 generations. Beyond 30 generations, the external crossvalidation error starts to increase, indicating that overfitting starts to occur after 30 generations. The graphs show that the ECV error estimate is more pessimistic than the test error estimates. However, the shape of both error curves are quite identical.

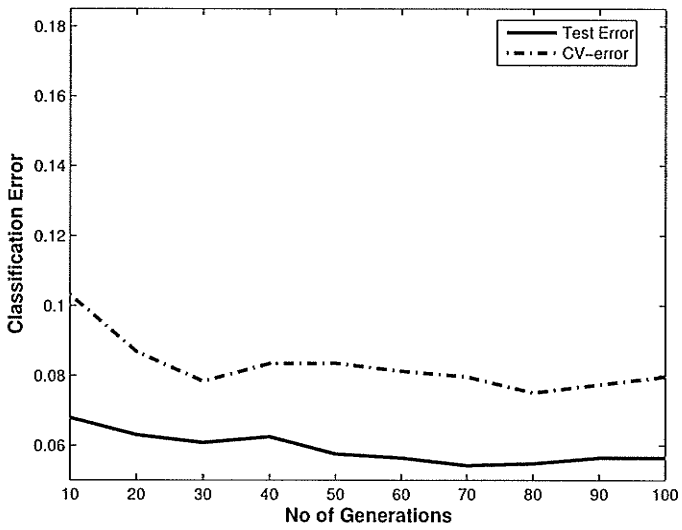

Figure 3.2: Classification error rates of the test for overfitting on Dataset 1 for increasing generations of the GA-ORS

Figures 3.4 and 3.5 illustrate the histogram of occurrence of the features selected from both datasets for the random seed initialization and the constant seed initialization. In Figures 3.4 and 3.5, the histograms on the left column are the feature frequencies of the subset selections initialized with random seeds, while the histograms on the right column are for the feature frequencies of subset selections initialized with the same constant

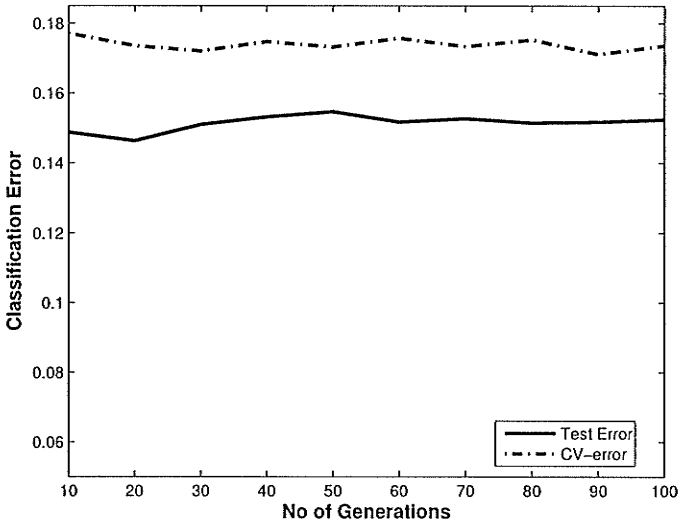

Figure 3.3: Classification error rates of the test for overfitting on Dataset 1 for increasing generations of the GA-ORS

seed. The histograms on the left column show that with different initialization seeds, the feature subsets selected by the GA-ORS vary. However, with the same initialization seed the same feature subset is selected by the GA-ORS from the same initialization seed. The histograms also show that the GA-ORS, like other sub-optimal wrapper-based feature selection algorithms, selects different feature subsets from different dataset partitions with the same parameters. Therefore, the results of these investigations are summarized as follows:

- The optimum number of generations that the GA-ORS will cycle through without overfitting is 30 generations for these datasets.
- The GA-ORS is unstable with respect to perturbations in the training data.
- Different initializations of the GA-ORS result in different feature subset selections.

Based on these conclusions, I present an evidence accumulation based feature selection strategy.

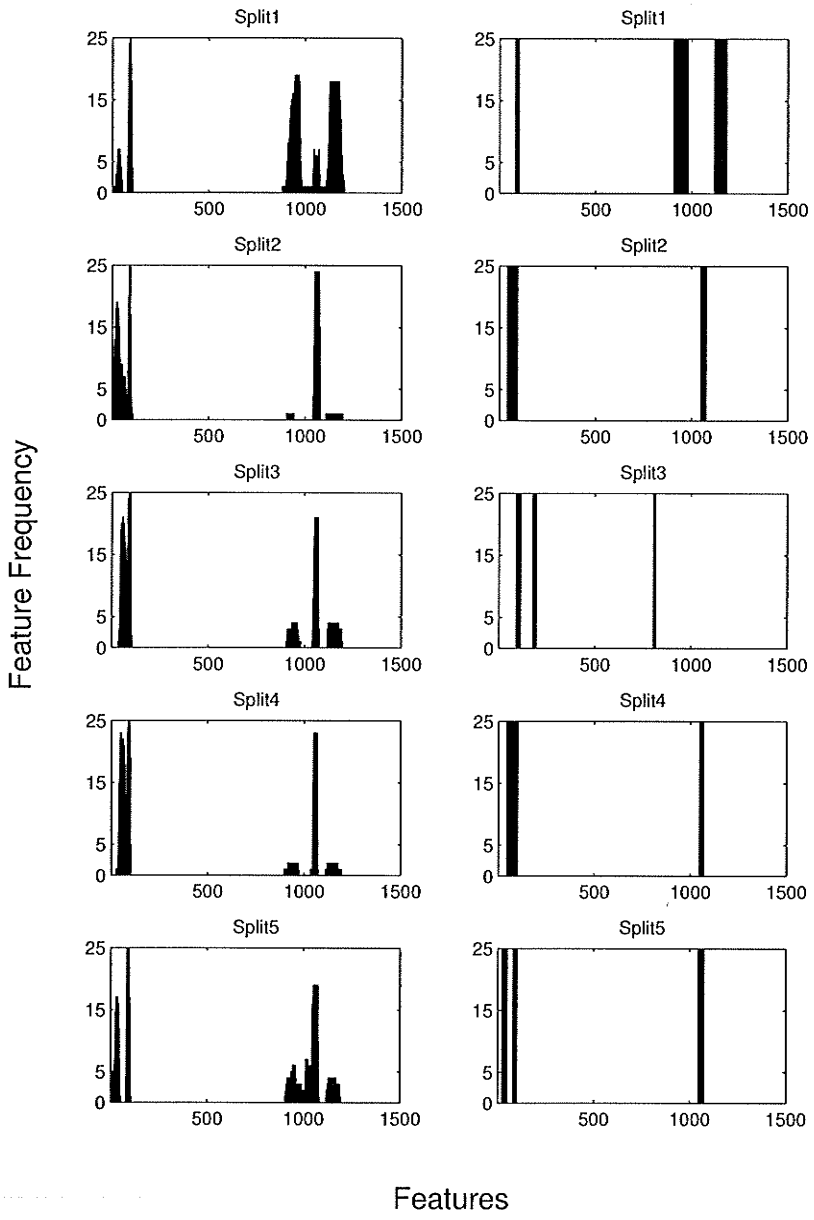

Figure 3.4: Feature frequency histograms of the 5 random splits of Dataset 1 used to test feature subset stability in the GA-ORS

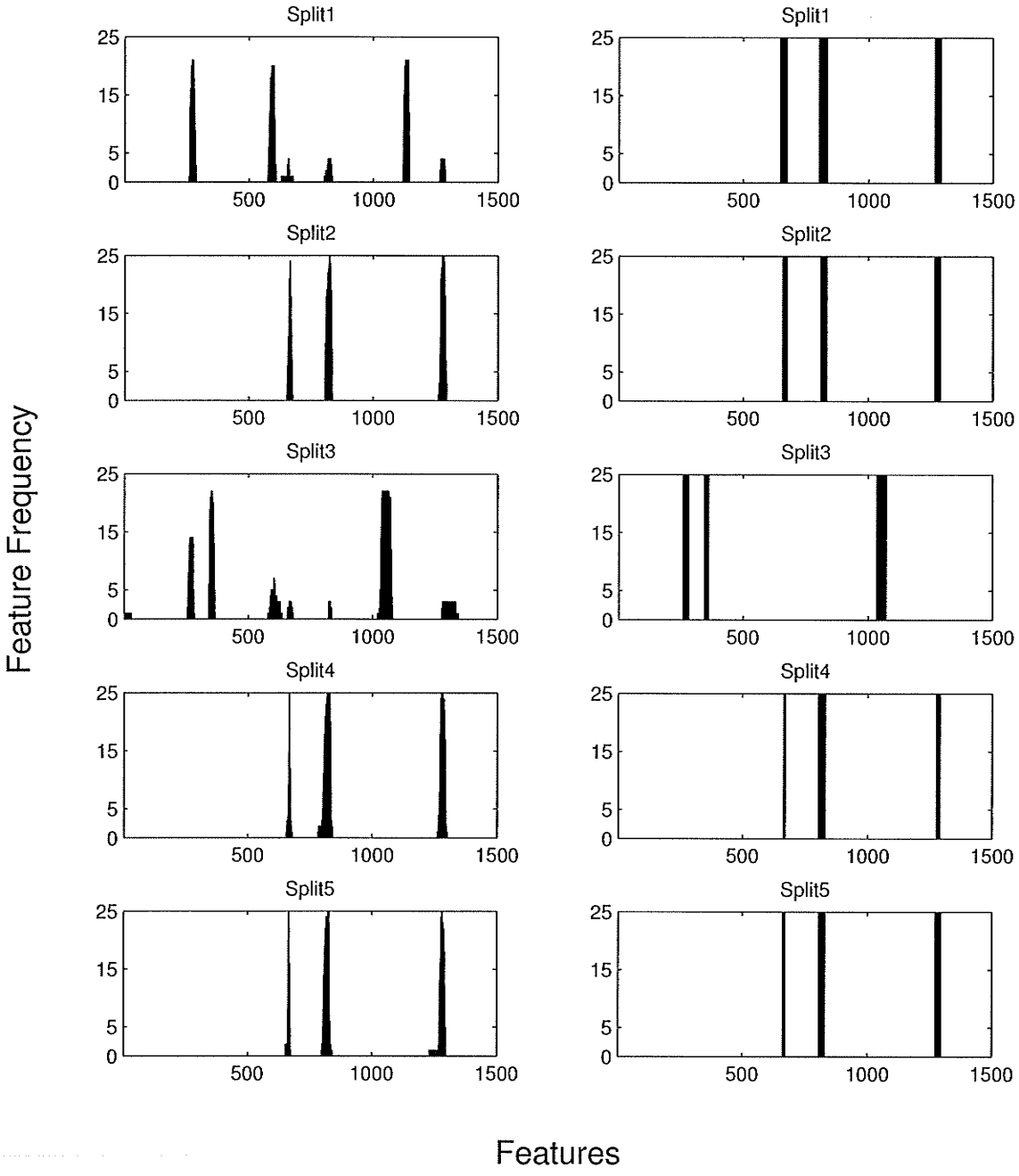

Figure 3.5: Feature frequency histograms of the 5 random splits of Dataset 2 used to test feature subset stability in the GA-ORS

3.4 Evidence Accumulation Based Feature Selection

The proposed evidence accumulation feature selection strategy (EAFS) is presented in this section. The EAFS combines independent evidence of the discriminatory features in the dataset to arrive at a final subset of discriminatory features from the combined evidence. The GA-ORS is used to collect the evidence, which is combined into a histogram of the frequency occurrence of the features in the generated evidence. The final feature subset selection is obtained from the histogram with a thresholding technique which is discussed in section 3.4.3. The methodology/experimental procedure of the EAFS is illustrated in the pseudocode in algorithm 1 and explained in the following subsections.

3.4.1 Collecting Evidence - Producing Discriminatory Feature Subsets

The evidence collection phase takes advantage of the subset instability observed in the GA-ORS when it is randomly initialized, and when the dataset partition is varied. Consequently, evidence is generated by first making multiple feature subset selections from the data using the GA-ORS. Each feature subset selection is initialized with a random seed value.

In the experiment, each dataset was partitioned into K ($k = 10$) random training set-test set splits (see stratification in Table 3.1). I used $N = 100$ different random seeds to carry out N feature extractions via the GA-ORS on each training set.

Algorithm 1 Evidence accumulation based feature selection algorithm

Input:

R = Number of regions to be selected

N = Maximum number of feature extractions

G = Number of generations

SF = Selection frequency

1. Generate K random splits of the dataset
2. for each random split
   2.1 For $i = 1$ to N
      2.1.1 Select initial random seed r
      2.1.2 Initialize GA-ORS with seed r and G then
         select R feature regions from split
      2.1.3 Increment the frequency count of the selected features
3. Plot Histogram of feature frequency counts
4. Select feature with SF selection frequency
5. Compute predictive accuracy of threshold SF features
   using external crossvalidation

3.4.2 Combining the Evidence - The Feature Frequency Histogram

The generated evidence is combined by accumulating the frequency occurrence of all features extracted during evidence generation into feature frequency histogram. Figure 3.6 is an example of a feature frequency histogram. I assume that the more frequently a

feature is selected, the more likely it is to be important. In the experiment, I combined the features extracted in the $N = 100$ GA-ORS runs into a feature frequency histogram.

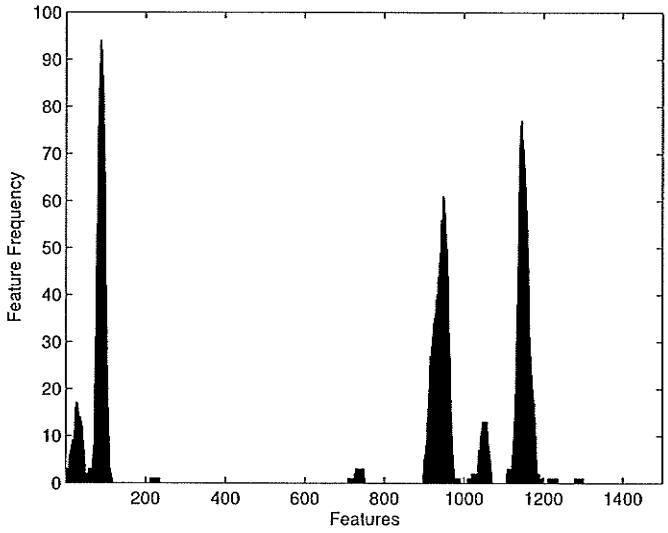

Figure 3.6: An example of a feature frequency histogram

3.4.3 Extracting the Final Feature Subset from the Combined Evidence - the Selection Frequency

The final group of features is obtained by selecting features on the feature frequency histogram with frequencies greater than or equal to a user defined frequency fraction threshold. I refer to this frequency fraction threshold as the selection frequency (SF) of the feature groups. A 0.25SF identifies features occurring with a frequency of $0.25N$ or greater in N selections. In the experiments, I selected final features subsets at 0.25, 0.50, 0.75, and 1.0 SFs.

3.4.4 Comparing EAFS Results with Results Obtained on an SVM

The EAFS was evaluated using external crossvalidation. A low external crossvalidation error will signify that the proposed method generalizes well and that its results must be significant. The ECV error of the EAFS was computed as follows. First, the EAFS was used to select a single final feature subset from the training set of the K splits of each of the datasets. An LDA classifier is learned on the training set with the selected feature subset. The resulting classifier is used to classify the samples in the corresponding split test set. The classification error, the proportion of the incorrectly classified test samples, is recorded. The external cross validation is calculated as the average classification error and standard deviation over the K splits.

The ECV error of the EAFS was benchmarked with the results obtainable with the dataset on an SVM. SVMs are state-of-the-art classifiers that generalize well on high-dimensional datasets with small sample sizes [26]. SVMs use L_2-norm regularization to overcome the overfitting problem. The LIBSVM [11] was used in this work. The training portion of each of the K dataset splits was used to learn SVM classifiers which were subsequently used to classify the corresponding test set samples of the $k = 10$ dataset splits.

3.4.5 Results of the Experimental Tests on the EAFS

The results of the experimental tests of the EAFS on Datasets 1 and 2 are presented in Tables 3.2 and 3.3. The summary of the results in Tables 3.2 and 3.3 are presented as graphs in Figures 3.7 and 3.8.

Tables 3.2 and 3.3 present the classification errors of the feature subsets selected with the EAFS for each of the ten random splits of each dataset. The results are presented for 10 to 100 generations of the GA-ORS and SF values 0.25, 0.50, 0.75, and 1.00. The results show that as the SF values for selecting the final feature subsets increase, the classification error increases. The error increase may be attributed to the decrease in the number of extracted final features as the SF increase. High SF values tend to filter out more good features. The results show that only on few occasions did features that satisfy the $1.00SF$ occur, and even such feature regions had considerably high classification errors. The results show that features selected at $0.25N$ have lower classification errors than features selected at higher SFs.

The graphs in Figures 3.7 and 3.8 show that the classification error of selected features decrease as the number of generations increase from 10 to 30. However, as the number of generations increase from 30 to 100, the classification error increased. This observation suggests that as the number of generations increases, the feature selection procedure starts to adapt to the peculiarities of the training set and that overfitting commences beyond 30 generations.

I compare the EAFS results with the results obtained using the SVM benchmark (see Table 3.4). The external crossvalidation errors of the SVM on Dataset 1 and Dataset 2 are 0.053 ± 0.028 and 0.176 ± 0.326 respectively. The EAFS results compare well with these results. This comparison is an indication that the EAFS was able to obtain interpretable candidate biomarkers without sacrificing classification accuracy.

Table 3.2: Classification errors on split test sets for feature subsets selected with EAFS from the 10 random splits of Dataset 1

G	SF	Split1	Split2	Split3	Split4	Split5	Split6	Split7	Split8	Split9	Split10	Av.Error
10	25	0.14	0.05	0.07	0.07	0.04	0.11	0.12	0.08	0.07	0.03	0.08
	50	0.18	0.07	0.11	0.08	0.05	0.14	0.16	0.08	0.10	0.04	0.10
	75	0.29	0.11	0.14	0.45	0.08	0.42	0.41	0.15	0.27	0.10	0.24
	100	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR	
20	25	0.08	0.08	0.05	0.00	0.04	0.10	0.10	0.08	0.07	0.04	0.06
	50	0.12	0.08	0.07	0.00	0.07	0.14	0.12	0.08	0.07	0.04	0.08
	75	0.14	0.08	0.07	0.10	0.07	0.42	0.15	0.08	0.07	0.10	0.13
	100	NR	0.12	0.44	NR	NR	NR	NR	NR	NR	NR	0.28
30	25	0.08	0.08	0.05	0.04	0.04	0.10	0.10	0.08	0.05	0.04	0.07
	50	0.10	0.07	0.07	0.08	0.07	0.12	0.12	0.10	0.05	0.04	0.08
	75	0.12	0.07	0.14	0.45	0.07	0.41	0.12	0.10	0.07	0.04	0.16
	100	NR	0.21	0.42	NR	0.47	NR	NR	0.16	NR	NR	0.32
40	25	0.08	0.05	0.05	0.03	0.05	0.10	0.08	0.08	0.05	0.16	0.08
	50	0.08	0.07	0.11	0.08	0.05	0.41	0.12	0.10	0.05	0.04	0.11
	75	0.10	0.12	0.44	0.08	0.07	0.42	0.12	0.08	0.05	0.04	0.15
	100	NR	0.12	0.42	NR	0.07	NR	NR	0.18	NR	NR	0.20
50	25	0.08	0.14	0.05	0.07	0.05	0.11	0.08	0.10	0.05	0.16	0.09
	50	0.08	0.08	0.10	0.10	0.05	0.41	0.08	0.10	0.05	0.04	0.11
	75	0.10	0.12	0.44	0.10	0.07	0.42	0.12	0.10	0.05	0.04	0.16
	100	NR	0.12	0.42	NR	0.05	NR	NR	0.18	NR	0.48	0.25
60	25	0.08	0.14	0.05	0.04	0.05	0.11	0.08	0.08	0.05	0.16	0.09
	50	0.08	0.08	0.10	0.10	0.05	0.29	0.08	0.08	0.05	0.04	0.10
	75	0.10	0.12	0.44	0.44	0.05	0.42	0.12	0.10	0.05	0.04	0.19
	100	0.47	0.12	NR	NR	NR	0.41	NR	0.18	NR	0.04	0.24
70	25	0.08	0.14	0.11	0.03	0.05	0.11	0.08	0.08	0.05	0.18	0.09
	50	0.08	0.08	0.10	0.10	0.07	0.11	0.08	0.08	0.05	0.16	0.09
	75	0.08	0.12	0.21	0.11	0.05	0.42	0.10	0.10	0.05	0.04	0.13
	100	0.47	0.12	NR	NR	0.45	NR	NR	0.18	NR	0.04	0.25
80	25	0.08	0.14	0.05	0.15	0.05	0.11	0.08	0.07	0.04	0.16	0.09
	50	0.08	0.07	0.10	0.10	0.05	0.11	0.08	0.10	0.05	0.16	0.09
	75	0.08	0.12	0.44	0.08	0.05	0.42	0.10	0.11	0.05	0.04	0.15
	100	0.47	0.12	NR	NR	NR	NR	NR	0.18	NR	0.47	0.31
90	25	0.08	0.07	0.05	0.07	0.05	0.11	0.08	0.10	0.04	0.18	0.08
	50	0.08	0.07	0.10	0.07	0.07	0.12	0.08	0.11	0.04	0.16	0.09
	75	0.08	0.12	0.44	0.08	0.05	0.42	0.10	0.11	0.05	0.04	0.15
	100	0.47	0.12	NR	NR	NR	NR	0.37	0.12	NR	0.04	0.22
100	25	0.08	0.14	0.11	0.14	0.05	0.11	0.08	0.08	0.04	0.18	0.10
	50	0.08	0.07	0.10	0.10	0.07	0.12	0.08	0.11	0.04	0.04	0.08
	75	0.08	0.10	0.22	0.08	0.05	0.42	0.10	0.11	0.05	0.04	0.13
	100	0.48	0.21	NR	NR	NR	NR	0.37	0.12	NR	0.04	0.24

NR - No Regions at the specified frequency threshold *G - Generations* *SF - Selection Frequency*

Table 3.3: Classification errors on split test sets for feature subsets selected with EAFS from the 10 random splits of Dataset 2

G	SF	Split1	Split2	Split3	Split4	Split5	Split6	Split7	Split8	Split9	Split10	Av.Error
10	25	0.12	0.12	0.16	0.11	0.13	0.14	0.13	0.18	0.20	0.14	0.14
	50	0.12	0.21	0.16	0.12	0.13	0.13	0.12	0.21	0.20	0.15	0.16
	75	NR	0.22	0.21	0.13	0.10	0.13	0.18	0.34	0.20	0.17	0.19
	100	NR	NR	NR	NR	NR	NR	NR	NR	NR	NR	-
20	25	0.13	0.12	0.16	0.11	0.10	0.12	0.13	0.18	0.18	0.16	0.14
	50	0.18	0.15	0.16	0.13	0.11	0.14	0.13	0.18	0.16	0.15	0.15
	75	NR	0.13	0.23	0.13	0.12	0.14	0.13	0.18	0.20	0.15	0.16
	100	NR	NR	NR	0.34	0.20	0.23	0.23	0.32	NR	0.38	0.28
30	25	0.19	0.13	0.18	0.12	0.12	0.14	0.13	0.18	0.16	0.16	0.15
	50	0.18	0.14	0.21	0.12	0.12	0.14	0.13	0.18	0.18	0.16	0.15
	75	0.18	0.13	0.20	0.12	0.12	0.14	0.13	0.18	0.18	0.16	0.15
	100	NR	NR	NR	0.17	0.11	0.23	0.23	0.32	0.22	0.38	0.24
40	25	0.18	0.13	0.20	0.12	0.12	0.14	0.13	0.18	0.17	0.15	0.15
	50	0.18	0.14	0.19	0.11	0.12	0.14	0.13	0.18	0.17	0.16	0.15
	75	0.18	0.13	0.21	0.12	0.12	0.14	0.13	0.18	0.18	0.15	0.15
	100	NR	NR	NR	0.20	0.15	0.14	0.15	0.29	0.22	0.38	0.22
50	25	0.18	0.15	0.19	0.12	0.12	0.15	0.13	0.18	0.16	0.16	0.15
	50	0.18	0.15	0.19	0.12	0.12	0.13	0.13	0.18	0.18	0.15	0.15
	75	0.18	0.14	0.20	0.11	0.12	0.13	0.13	0.18	0.19	0.16	0.15
	100	NR	0.39	NR	0.17	0.20	0.14	0.15	0.30	0.21	0.38	0.24
60	25	0.18	0.15	0.19	0.12	0.12	0.15	0.13	0.17	0.17	0.16	0.15
	50	0.18	0.15	0.19	0.12	0.12	0.15	0.12	0.18	0.17	0.15	0.15
	75	0.18	0.14	0.19	0.12	0.12	0.13	0.13	0.18	0.19	0.16	0.15
	100	NR	NR	NR	0.34	0.33	0.23	0.15	0.31	0.21	0.38	0.28
70	25	0.19	0.15	0.19	0.12	0.12	0.15	0.13	0.17	0.18	0.17	0.16
	50	0.18	0.15	0.19	0.12	0.12	0.14	0.13	0.18	0.15	0.16	0.15
	75	0.18	0.15	0.19	0.12	0.12	0.14	0.13	0.19	0.16	0.16	0.15
	100	NR	NR	NR	0.17	0.20	0.19	0.15	0.30	0.21	0.39	0.23
80	25	0.18	0.15	0.19	0.12	0.12	0.14	0.13	0.17	0.16	0.17	0.15
	50	0.18	0.15	0.19	0.10	0.12	0.14	0.12	0.17	0.15	0.15	0.15
	75	0.19	0.15	0.19	0.12	0.12	0.14	0.12	0.18	0.15	0.16	0.15
	100	NR	NR	NR	0.30	0.20	0.19	0.14	0.32	0.20	0.39	0.25
90	25	0.18	0.15	0.15	0.12	0.12	0.15	0.13	0.17	0.18	0.16	0.15
	50	0.18	0.15	0.19	0.12	0.12	0.13	0.12	0.17	0.15	0.15	0.15
	75	0.18	0.15	NR	0.11	0.12	0.14	0.12	0.17	0.15	0.15	0.14
	100	NR	NR	NR	0.18	NR	0.19	0.15	0.31	0.20	0.39	0.24
100	25	0.18	0.15	0.19	0.12	0.12	0.15	0.13	0.17	0.19	0.16	0.15
	50	0.19	0.15	0.19	0.12	0.11	0.13	0.12	0.17	0.15	0.15	0.15
	75	0.18	0.14	0.19	0.12	0.12	0.14	0.12	0.17	0.15	0.13	0.15
	100	NR	NR	NR	0.18	0.19	0.19	0.20	0.32	0.20	0.38	0.24

NR - No Regions at the specified frequency threshold *G - Generations* *SF - Selection Frequency*

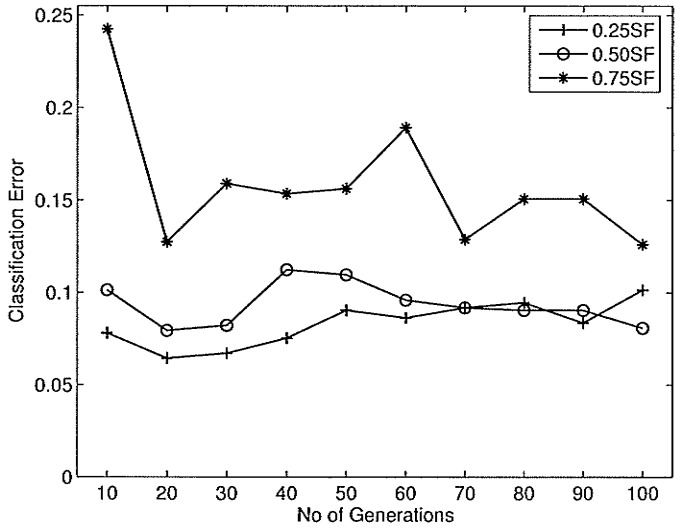

Figure 3.7: Average Classification error over the 10 random splits of Dataset1 for 10 to 100 generations

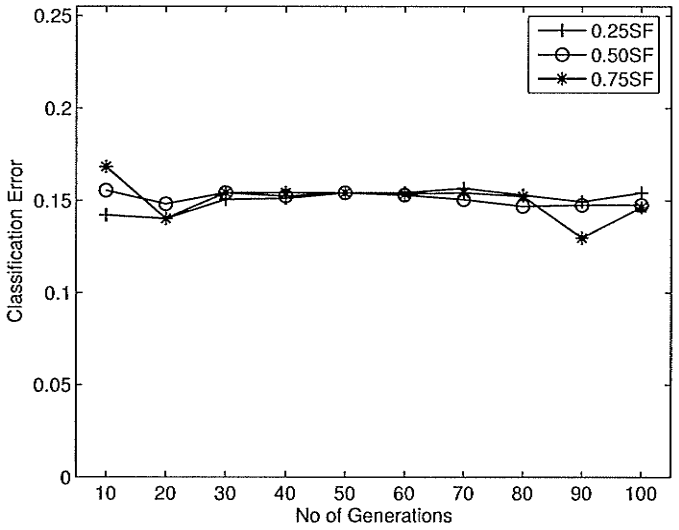

Figure 3.8: Average Classification error over the 10 random splits of Dataset2 for 10 to 100 generations

Table 3.4: Classification errors of random splits of Dataset 1 and Dataset 2 on SVM

Dataset1		*Dataset2*	
Random Splits	Misclassification Error	Random Splits	Misclassification Error
1	0.0958904	1	0.140244
2	0.0684932	2	0.176829
3	0.0547945	3	0.189024
4	0.0273973	4	0.115854
5	0.0547945	5	0.176829
6	0.0958904	6	0.189024
7	0.136986	7	0.20122
8	0.0273973	8	0.152439
9	0.0547945	9	0.182927
10	0.0410959	10	0.231707
Average Error ± Standard Dev.	**0.0534247±0.0277376**		**0.17561±0.326218**

Stability of features selected by the EAFS

The stability of the EAFS was assessed on the basis of consistency of the feature regions selected across the different dataset splits. Heatmaps (see Figures 3.9 and 3.10) were used to visualize and compare the feature regions selected by the EAFS from each of the 10 splits of both datasets at 0.25 SF. The heatmaps show that a significant proportion of the datasets agree on a consistent region of features. However, the strength of their agreement, which is indicated by the brightness of the region, differs from one dataset split to another.

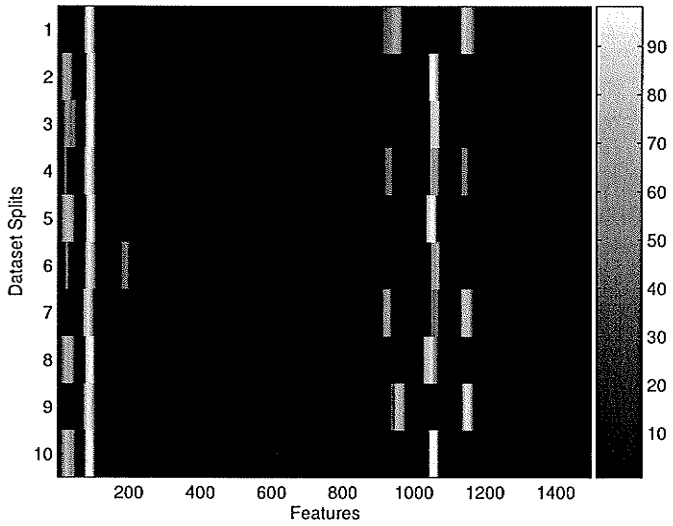

Figure 3.9: Heatmap of features selected by the EAFS from random splits of Dataset 1

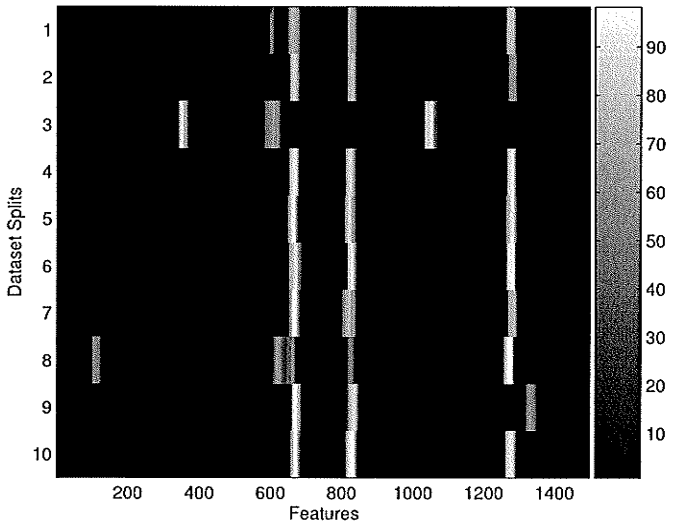

Figure 3.10: Heatmap of features selected by the EAFS from random splits of Dataset 2

3.4.6 Investigating the Effect of the Number of Random Feature Selections on the EAFS Evidence Generation

In the EAFS experiments discussed previously, the evidence generation procedure was based on 100 randomly initialized feature subset selections of the GA-ORS. In this sec-

tion, I investigate the effect of the number of random feature selections on the features selected by the evidence accumulation based feature selection strategy. The goal of this investigation is to determine if fewer feature subsets will suffice for evidence generation. The 10 data splits used in the EAFS experiment in section 3.4 were used in this investigation. I made 10 to 100 (in steps of 10) randomly initialized feature selections from each of the training portions using the GA-ORS. Three (3) feature regions were selected and the number of generations was set to 30. Features that occur at the 0.25 SF for each random selection were selected and the selected features were evaluated on the corresponding 10 test sets. Figure 3.11 and Figure 3.12 illustrate the results obtained from this investigation for datasets 1 and 2. The results show that the external cross validation error obtained for the random selections did not vary with respect to the number of selections. Therefore, a smaller number of random feature selections should suffice for evidence generation.

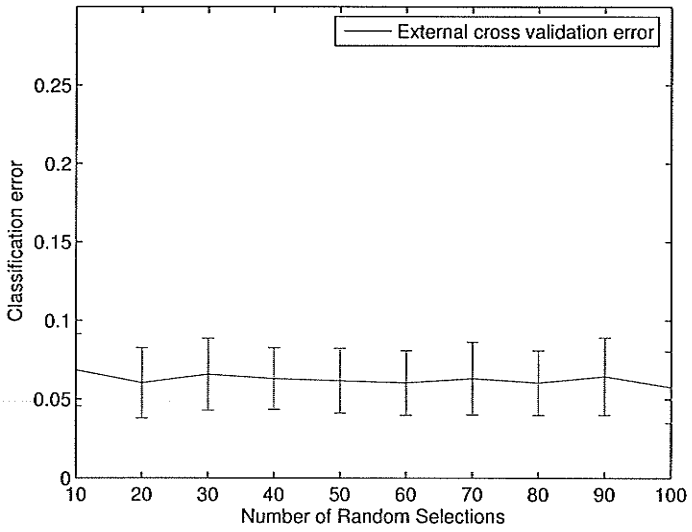

Figure 3.11: Testing the effect of the number of random selections on Dataset 1

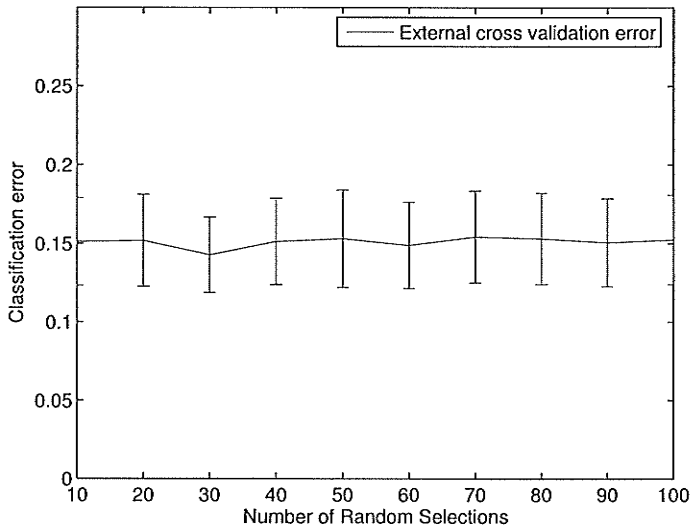

Figure 3.12: Testing the effect of the number of random selections on Dataset 2

Summary

The evidence accumulation feature selection strategy was presented in this Chapter. The GA-ORS was used to generate evidence of possible biomarkers in two real life spectral datasets. The evidence generation involved multiple feature subset selections from the dataset. Each selection is randomly initialized. The generated evidence was combined into a feature frequency histogram. A single feature subset was selected from the histogram by selecting features in the histogram that occur with a frequency value greater than or equal to a user defined threshold value referred to as the selection frequency.

The results obtained from experiments on the EAFS with two real life datasets show that the strategy is capable of obtaining feature regions with a classification error similar to that obtainable on a SVM. The low ECV error of the EAFS signifies that the feature regions are potential biomarkers. In the next chapter, an MCS based feature selection

strategy that incorporated the EAFS into an RSM is presented.

Chapter 4

Random Subspace Feature Selection

4.1 Introduction

In this chapter, an MCS-based feature selection strategy that incorporates the EAFS into the random subspace method (RSM) is presented. The RSM is an MCS that generates its classifier ensemble from random subspaces of the dataset feature space. As a result, the SFR of the dataset partitions used to create the classifier ensembles is better than the entire dataset; the data dimensionality reduces, while the number of samples remain the same. In [46] Skurichina show that RSMs perform well on datasets with collections of redundant discriminatory features that are distributed almost uniformly across the feature space. This RSM feature makes it well suited for analyzing biomedical spectra, which have many redundant features [37]. The RSM-FS also extends the RSM via a noise addition technique for identifying the biomarkers in the dataset.

4.2 Random Subspace Feature Selection

In this section, I discuss the Random Subspace Method Feature Selection (RSM-FS) algorithm. The RSM-FS incorporates the evidence accumulation based feature selection strategy presented in Chapter 3 into the random subspace method. The RSM-FS strategy is illustrated in algorithm 2, and discussed as follows.

The RSM-FS creates the multiple classifiers required by an RSM by dividing the feature space of the dataset into contiguous feature subspaces. That is, given a biomedical dataset L consisting N samples of the form $(x_1, y_1), ..., (x_N, y_N)$ where x_i is a d-dimensional vector of features of sample i, and y_i is the output label of sample i signifying the class or type of the sample (e.g. diseased or healthy). The feature vector x_i is divided into non-overlapping contiguous feature subspaces that are approximately equal in size. That is, given that feature vector x_i consists of features $x_{i1}, x_{i2}, ..., x_{id}$, if $1 < a < b < d$, then the following are contiguous partitions of x_i : $[x_{i1}, x_{i2}, ..., x_{ia}]$, $[x_{i(a+1)}, x_{i(a+2)}, ..., x_{ib}]$ and $[x_{i(b+1)}, x_{i(b+2)}, ..., x_{id}]$. These feature subspaces serve as the diverse data partitions required for a successful RSM. The EAFS is used to select a feature subset from each feature subset space. The resulting feature subset is used to learn a linear discriminant classifier (LDA). The LDAs from each of the feature subspaces are combined into an RSM classifier via a majority vote. Dividing the feature space into smaller subspaces will reduce the data partition dimensionality relative to its sample size. Therefore, the feature subsets selected by the GA-ORS should be better than the selections made over the entire feature subspace.

Algorithm 2 Random subspace based feature selection algorithm

Input:

R = Number of regions to be selected

N = Maximum number of feature extractions

G = Number of generations

SF = Selection frequency

F = Number of feature subspace splits

1. Generate K random splits of the dataset

2. for each random split

2.1 Divide the feature space F equal feature subspaces

2.1.1 For each feature subspace

2.1.1.1 Use the EAFS (using parameters R,N,G,SF)

to select feature subsets from the split training set

2.1.1.2 Learn a classifier on the feature subset

selected by the EAFS

2.1.2 Combine the classifiers of each feature subspace into an RSM

2.1.3 Classify the split test set samples using the RSM

4.2.1 Optimum Number of Feature Subspace Splits

An important requirement for the performance of an RSM is that each feature subspace must contain discriminatory features [46]. If the discriminatory features are concentrated in only a few feature subspaces, poor classifiers will be constructed from the subspaces

void of discriminatory features. These poor classifiers will in turn adversely affect the performance of the RSM. In this section, I investigated the optimum number of feature subspaces required for the RSM-FS in this study. In this investigation, the feature subspace of each dataset training portion was partitioned into the following approximately equal feature subspaces - 3, 5, 7, 9, 11, and 15. The EAFS was used to select feature subsets from each feature subspace using the following parameters : 30 generations; 15 random feature selections; and a selection frequency of $0.25SF$. The feature subsets selected from the feature subspace were used to learn RSM classifier ensemble using LDA. The RSM was subsequently used to classify the samples in the test set using a majority vote.

Results of Test for Optimum Number of Feature Subspace Splits

The results for the optimum number of feature subspace splits of Dataset 1 and Dataset 2 are presented in Figures 4.1 and 4.2 respectively. The graphs show an increase in the RSM error as the number of splits increase. This increase is an indication that as the feature subspace size decreases, the feature selection strategy starts to overfit by selecting irrelevant features. Consequently, smaller feature subspace sizes appear to be more susceptible to overfitting than larger feature subspace sizes. Three (3) feature subspaces were used in the experiments in subsequent sections.

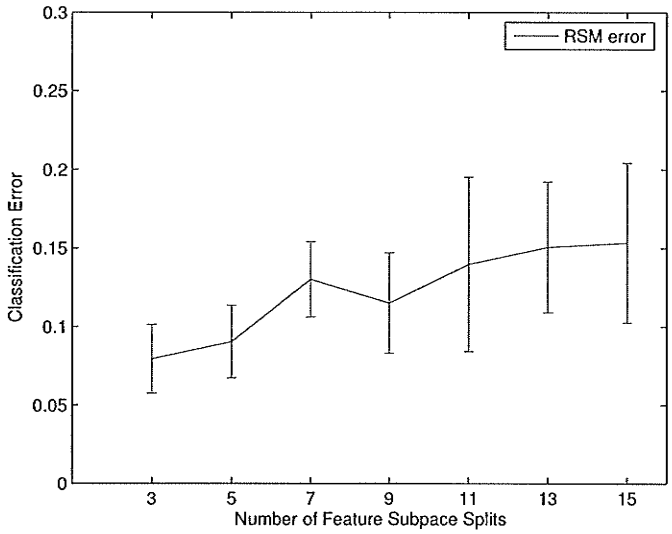

Figure 4.1: Testing for the optimum number of feature subspace splits

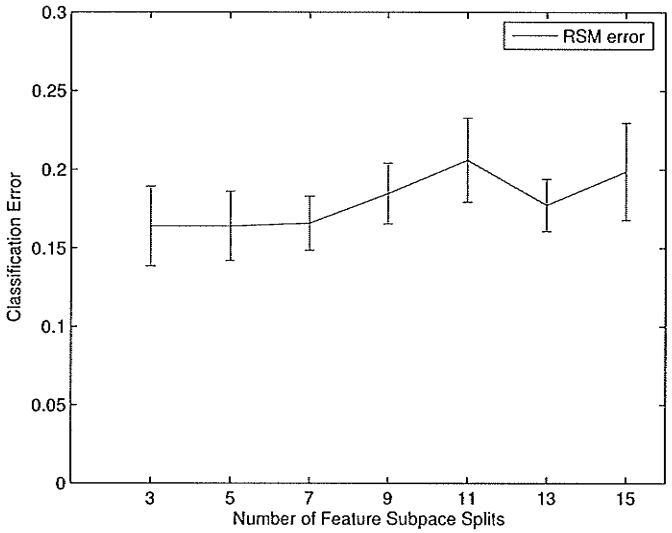

Figure 4.2: Testing for the optimum number of feature subspace splits

4.2.2 Experimental Test on the RSM-FS

The ten (10) dataset splits of Dataset 1 and Dataset 2 used in the EAFS experiments in Section 3.4 were used in this experiment. Each feature space, composed of 1500

features, was split into 3 contiguous feature subspaces each containing 500 features. Consequently, the training set of each dataset split was divided into 3 partitions, each containing the same number of samples as the original dataset, and a dataset dimension of 500. The EAFS was used to select feature regions from each of the 3 partitions. The parameters used to run the EAFS are: the number of regions to be selected was 3, the maximum number of feature extractions was 15, the number of generations was 30, and the selection frequency was 0.25 SF. The feature regions selected from each of the 3 training set partitions were used to learn LDA classifiers that were combined into an RSM. The resulting RSM was used to classify samples in the corresponding test set of each dataset split.

4.2.3 Evaluating RSM Features for Possible Biomarkers

An important objective of this research is to select possible biomarkers that can serve as a basis for further research to a domain specialist. To test if the RSM-FS actually selects possible biomarkers, I used the following strategy proposed by Breiman [10] to evaluate the relevance of the features in the RSM on the accuracy of the RSM. This strategy uses noise to evaluate the relevance of the constituent features of a MCS on the performance of the MCS. After constructing the RSM-FS as described in Section 4.2, each feature region is evaluated by apply noise to each feature region in the test portion of the dataset split. The RSM is used to classify the "noised out" data, and the difference between the RSM error before noising out values of the feature region, and the RSM error after the feature region is noised out serves an indication of the relevance of the feature region in

the RSM.

4.2.4 Noising Out the Feature Regions

The procedure for noising out feature regions in this experiment is in [9], and is as follows. First, the standard deviation of the test set values of the feature region is computed. The standard deviation is calculated a second time with only values less than 2.5 times the initial standard deviation value from the feature region mean value. The noised out feature region is then generated as follows:

$$R_i^* = R_i + Z_i.sd, i = 1, ..., N$$

where R_i is the feature region value for sample i, sd is the standard deviation, and Z_i are independent unit normals.

4.2.5 Result of the RSM-FS Experiments

Tables 4.1 and 4.2 show the results of the experiments with the RSM-FS on the 10 splits of dataset 1 and dataset 2 respectively, and the importance values of the feature regions of splits' on the RSMs. The results show that show that more than half of the feature regions in each of the RSM classifiers resulted in an RSM error increase of approximately 10% or more. It is assumed that these feature regions have a significant influence on the performance of their RSMs if noising them out results in the increase in RSM error. The results also show similar regions appear to show significant influence on the RSMs' performance across the splits. For example in Dataset 1, feature region 1001-1016 (or

subsets of this region) show significant influence in all the 10 splits of the dataset as a result of the noise testing. In Dataset 2 also, feature region 821-843 (or subsets of this region) shows significant influence in almost all the splits (except splits 4 and 8) as a result of the noise testing. These results show that the RSM-FS strategy is selecting possible biomarkers.

Table 4.1: Feature regions selected by the RSM-FS from dataset 1 and their importance values

	RSM Error (b)														
Split1	**0.110**	32-34	85-97	135-138	140-142	274-275	278-282	286-295	730-745	804-815	938-951	1001-1009	1043-1060	1132-1143	
a_1		0.123	0.206	0.110	0.301	0.301	0.301	0.301	0.164	0.164	0.164	0.274	0.274	0.274	
a_1-**b**		0.014	0.096	0.000	0.192	0.192	0.192	0.192	0.055	0.055	0.055	0.164	0.164	0.164	
Split2	**0.055**	26-34	86-101	273-281	283-291	735-750	801-814	936-951	1001-1011	1044-1059	1135-1148				
a_2		0.096	0.123	0.110	0.206	0.110	0.110	0.110	0.151	0.151	0.151				
a_2-**b**		0.041	0.069	0.055	0.151	0.055	0.055	0.055	0.096	0.096	0.096				
Split3	**0.096**	38-40	85-94	96	139-149	286-294	736-746	805-815	942-953	1001-1011	1052-1063	1133-1144			
a_3		0.123	0.151	0.123	0.219	0.219	0.192	0.192	0.192	0.233	0.233	0.233			
a_3-**b**		0.027	0.055	0.027	0.123	0.123	0.096	0.096	0.096	0.137	0.137	0.137			
Split4	**0.082**	84-103	174-182	280-281	306	308-313	731-744	802-812	814	942-951	1054-1065	1447-1459	1471-1478	1485-1495	
a_4		0.110	0.123	0.123	0.233	0.233	0.219	0.219	0.219	0.219	0.151	0.151	0.151	0.151	
a_4-**b**		0.027	0.041	0.041	0.151	0.151	0.137	0.137	0.137	0.137	0.069	0.069	0.069	0.069	
Split5	**0.041**	34-35	37-41	86-97	142-148	271-274	278-282	286-291	736-744	804-815	941-950	1001-1016	1046-1058	1139-1144	1278-1285
a_5		0.069	0.096	0.069	0.192	0.192	0.192	0.192	0.096	0.096	0.096	0.164	0.164	0.164	0.164
a_5-**b**		0.027	0.055	0.027	0.151	0.151	0.151	0.151	0.055	0.055	0.055	0.123	0.123	0.123	0.123
Split6	**0.082**	38-40	84-98	100-101	176-184	277-284	310-318	732-744	797-812	941-954	1001-1009	1050-1062	1132-1143		
a_6		0.082	0.137	0.110	0.206	0.206	0.206	0.164	0.164	0.164	0.123	0.123	0.123		
a_1-**b**		0.000	0.055	0.027	0.123	0.123	0.123	0.082	0.082	0.082	0.041	0.041	0.041		
Split7	**0.082**	3-17	31-38	81-95	268-273	280-284	733-748	801-816	939-951	1001-1012	1055-1064	1129-1142			
a_7		0.096	0.110	0.110	0.260	0.260	0.206	0.206	0.206	0.247	0.247	0.247			
a_7-**b**		0.014	0.027	0.027	0.178	0.178	0.123	0.123	0.123	0.164	0.164	0.164			

b is the RSM error of each dataset split. a_i is the RSM error with noise in each feature region of split i. a_i-b is the importance of the feature region in split i

Table 4.1 – continued from previous page

Split8	**0.096**	30-40	87-100	357-366	730-744	802-803	807-810	812-817	941-954	1001-1014	1042-1058	1276-1293
a_8		0.110	0.123	0.137	0.233	0.233	0.233	0.233	0.233	0.233	0.233	0.233
a_8**-b**		0.014	0.027	0.041	0.137	0.137	0.137	0.137	0.137	0.137	0.137	0.137
Split9	**0.082**	22-38	84-98	271-279	282	734-746	799-809	938-956	1001-1009	1051-1063	1133-1144	
a_9		0.082	0.123	0.123	0.206	0.164	0.164	0.164	0.178	0.178	0.178	
a_9**-b**		0.000	0.041	0.041	0.123	0.082	0.082	0.082	0.096	0.096	0.096	
Split10	**0.055**	27-41	85-99	270-288	734-748	795-809	941-951	1001-1013	1052-1062	1279-1290		
a_{10}		0.069	0.082	0.096	0.192	0.192	0.192	0.178	0.178	0.178		
a_{10}**-b**		0.014	0.027	0.041	0.137	0.137	0.137	0.123	0.123	0.123		

b is the RSM error of each dataset split. a_i **is the RSM error with noise in each feature region of split** i**.** a_i**-b is the importance of the feature region in split** i

Table 4.2: Feature regions selected by the RSM-FS from dataset 2 and their importance values

	RSM Error (b)												
Split1	0.177	21-29	33-43	251-261	**667-686**	**828-839**	**907-920**	**1034-1047**	**1129-1132**	**1135-1137**	**1139-1142**	**1274-1287**	
a_1		0.177	0.165	0.213	**0.323**	**0.323**	**0.323**	**0.293**	**0.293**	**0.293**	**0.293**	**0.293**	
a_1-b		0.000	-0.012	0.037	**0.146**	**0.146**	**0.146**	**0.116**	**0.116**	**0.116**	**0.116**	**0.116**	
Split2	0.140	13-24	33-44	217-220	**222-223**	**225-229**	**663-673**	**825-838**	**916-927**	**1267-1283**	**1329-1341**	**1393-1401**	**1409-1413**
a_2		0.189	0.171	0.171	**0.262**	**0.262**	**0.281**	**0.281**	**0.281**	**0.317**	**0.317**	**0.317**	**0.317**
a_2-b		0.049	0.030	0.030	**0.122**	**0.122**	**0.140**	**0.140**	**0.140**	**0.177**	**0.177**	**0.177**	**0.177**
Split3	0.146	144-160	235-242	304-313	**332-351**	**663-675**	**824-838**	**918-929**	**961-969**	**1063-1076**	**1078-1087**	**1278-1286**	
a_3		0.146	0.159	0.171	**0.274**	**0.274**	**0.274**	**0.274**	**0.274**	**0.329**	**0.329**	**0.329**	
a_3-b		0.000	0.012	0.024	**0.128**	**0.128**	**0.128**	**0.128**	**0.128**	**0.183**	**0.183**	**0.183**	
Split4	0.165	17-31	35-43	83-85	**90**	**670-683**	**688-702**	**912-925**	**1206-1217**	**1275-1283**	**1377-1392**	**1398-1399**	**1405-1406**
a_4		0.195	0.195	0.213	**0.329**	**0.366**	**0.366**	**0.366**	**0.360**	**0.360**	**0.360**	**0.360**	**0.360**
a_4-b		0.030	0.030	0.049	**0.165**	**0.201**	**0.201**	**0.201**	**0.195**	**0.195**	**0.195**	**0.195**	**0.195**
Split5	0.128	26-27	35-42	**141-157**	**304-317**	**664-671**	**821-832**	**921-931**	**1153-1158**	**1271-1287**	**1294**	**1404-1408**	
a_5		0.140	0.159	**0.201**	**0.250**	**0.317**	**0.317**	**0.317**	**0.305**	**0.305**	**0.305**	**0.305**	
a_5-b		0.012	0.030	**0.073**	**0.122**	**0.189**	**0.189**	**0.189**	**0.177**	**0.177**	**0.177**	**0.177**	
Split6	0.171	16-29	31	33-43	**309-319**	**680-690**	**750-753**	**761-762**	**826-836**	**923-931**	**1039-1049**	**1266-1276**	**1404-1415**
a_6		0.177	0.183	0.213	**0.268**	**0.329**	**0.329**	**0.329**	**0.329**	**0.329**	**0.293**	**0.293**	**0.293**
a_6-b		0.006	0.012	0.043	**0.098**	**0.159**	**0.159**	**0.159**	**0.159**	**0.159**	**0.122**	**0.122**	**0.122**
Split7	0.171	11-47	254-259	263-264	**585-589**	**667-677**	**689-693**	**827-838**	**1048-1069**	**1103-1117**	**1272-1285**		
a_7		0.159	0.189	0.152	**0.366**	**0.366**	**0.366**	**0.366**	**0.354**	**0.354**	**0.354**		
a_7-b		-0.012	0.018	-0.018	**0.195**	**0.195**	**0.195**	**0.195**	**0.183**	**0.183**	**0.183**		
Split8	0.201	18-30	33-44	97-104	**629-645**	**731**	**738**	**741-747**	**752**	**755-761**	**1063-1075**	**1115-1133**	**1274-1286**

b is the RSM error of each dataset split. a_i is the RSM error with noise in each feature region of split i. a_i-b is the importance of the feature region in split i

Table 4.2 – continued from previous page

a_8		0.183	0.213	0.213	**0.354**	**0.354**	**0.354**	**0.354**	**0.354**	**0.354**	**0.354**	**0.354**	**0.354**
a_8-b		-0.018	0.012	0.012	**0.152**	**0.152**	**0.152**	**0.152**	**0.152**	**0.152**	**0.152**	**0.152**	**0.152**
Split9	0.177	149-162	273-312	649-670	**828-843**	**1215-1220**	**1273-1285**	**1339-1351**	**1394-1414**				
a_9		0.183	0.177	0.183	**0.354**	**0.366**	**0.366**	**0.366**	**0.366**				
a_9-b		0.006	0.000	0.006	**0.177**	**0.189**	**0.189**	**0.189**	**0.189**				
Split10	0.165	24-40	305-318	645-652	**665-675**	**770-777**	**828-840**	**1153-1162**	**1272-1281**	**1393-1402**	**1407-1409**		
a_{10}		0.171	0.195	0.201	**0.293**	**0.293**	**0.293**	**0.287**	**0.287**	**0.287**	**0.287**		
a_{10}-b		0.006	0.030	0.037	**0.128**	**0.128**	**0.128**	**0.122**	**0.122**	**0.122**	**0.122**		

b is the RSM error of each dataset split. a_i is the RSM error with noise in each feature region of split i. a_i-b is the importance of the feature region in split i

The external crossvalidation error of the RSM-FS was calculated from the ten splits of Dataset 1 and Dataset 2. The external crossvalidation error of the RSM-FS on Dataset 1 and Dataset 2 are 0.078 ± 0.021 and 0.164 ± 0.021 respectively. These results are comparable with the results obtained on the SVMs in Chapter 3. These results show that the RSM-FS generalizes well.

Heatmaps of the importance values of the dataset splits of Dataset 1 and Dataset 2 (see Figures 4.3 and 4.4) show that all ten splits of Dataset 1 have importance values of at least 10% on 5 regions : 735-752, 796-816, 937-954, 1001-1014 and 1041-1065. Similarly all ten splits of Dataset 2 have importance values of at least 10% on regions 1266-1290. Many other regions in Dataset 2 have importance values greater or equal to 10%. However, not all splits show such high importance values.

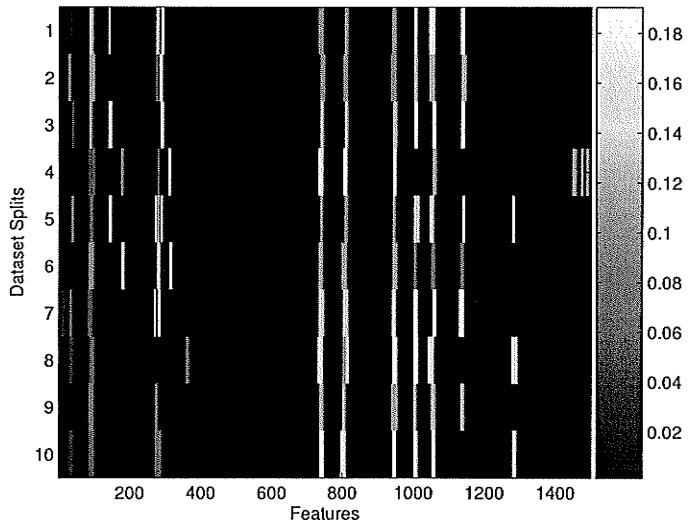

Figure 4.3: RSM Results for Dataset 1

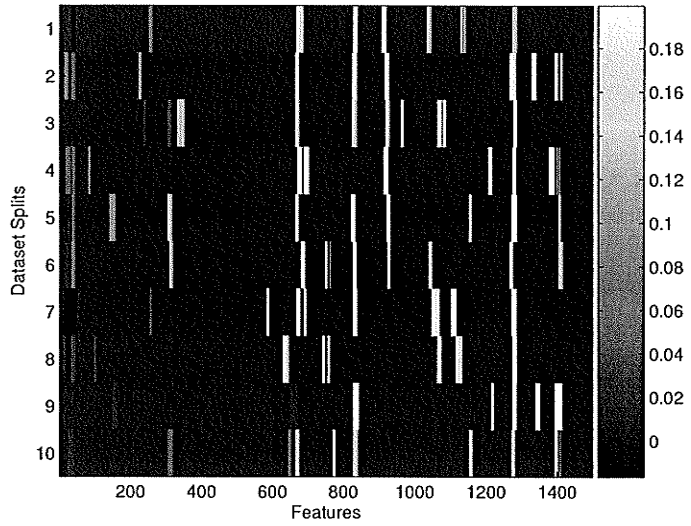

Figure 4.4: RSM Results for Dataset 2

4.2.6 Summary

The random subspace method based feature selection (RSM-FS) was presented in this chapter. The RSM-FS is a multiple classifier based feature selection strategy that incorporates the evidence accumulation base feature selection strategy into the random subspace method. The framework of the strategy was discussed in section 4.2. The experiments ran to evaluate the proposed strategy were discussed in Section 4.2.2 and the results of these experiments are in Section 4.2.5. A technique that uses noise to determine the importance of the features selected by the RSM-FS was discussed in Section 4.2.3. The results show that the RSM-FS selections feature subsets that have low classification errors and show almost similar degree of importance over all or most of the splits of the datasets.

Chapter 5

Conclusions and Recommendations

5.1 Conclusions

In this thesis a-multiple-classifier-system-based feature selection strategy for biomarker discovery, based on an evidence-accumulation-based feature selection (EAFS) strategy and the random subspace method, was presented. To the best of my knowledge, the EAFS is a novel attempt at applying evidence accumulation to supervised feature selection to arrive at a more robust selection. The MCS-based feature selection strategy uses an RSM in conjunction with the EAFS to select discriminatory features from data. The experimental results show that the proposed strategies discover relatively stable and important biomarkers from real-life datasets. These biomarkers can be included into disease profiles for further research by a domain expert.

The evidence accumulation feature selection takes advantage of the feature subset instability of wrapper-based feature selection algorithms on different dataset partitions

and/or different initialization parameters. The wrapper-based feature selection algorithm used in this thesis is GA-ORS [34]. The GA-ORS was used because it has the ability to select feature regions with varying sizes. Also it includes a preprocessing step that condenses these regions into even fewer features during feature evaluation. These features of the GA-ORS make it well-suited for biomarker discovery in high-dimensional biomedical data. The random subspace method feature selection (RSM-FS) strategy is the proposed MCS-based feature selection strategy presented in this thesis. As earlier mentioned, it is based on the EAFS and the RSM [25]. The importance of the feature regions selected by the RSM-FS is evaluated with a technique that applies noise to the feature regions in the test set, with the aim of evaluating the increase in the RSM error due to the application of the noise. An error increase serves to signify that the noised-out features are important and possible biomarkers.

The results obtained show that an evidence-accumulation-based approach can be successfully applied to supervised wrapper-based feature selection. In addition, the features selected perform well and compare well with results obtained on support vector machines. Also, the low external crossvalidation error of the EAFS signifies that the strategy does not overfit and that the selected features may be possible biomarkers. The RSM-FS performed well as a result of the smaller feature subspace partitions. Using the noise addition technique applied in this thesis, the results obtained show that features selected by the RSM-FS contained some important features, which with further research may turn out to be biomarkers for disease profiling.

5.2 Recommendations

In this thesis, I have been able to select possible biomarkers from two high-dimensional biomedical spectral datasets using the proposed RSM-FS strategy. Although these results are promising, the expertise of a domain expert is required to confirm the clinical applicability of these results.

The GA-ORS was used as the feature selection algorithm of choice. However, any wrapper-based feature selection algorithm should suffice. It will be interesting to compare the results obtained in this thesis with results obtained with the EAFS algorithm based on other wrapper based feature selection algorithms. Also, the evidence accumulation framework combined results from different initializations of a single feature selection algorithm. It will be interesting to explore the possibility of combining results obtained from the accumulation of results obtained from different wrapper based feature selection algorithms. Finally, It will be interesting to test the proposed strategy on a dataset with known biomarkers.

Bibliography

[1] C. Ambroise and G. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences in the United States of America*, 99(10):6562–6566, 2002.

[2] V. Aris and M. Recce. A method to improve detection of disease using selectively expressed genes in microarray data. In *Proceedings of the First Conference on Critical assessment of Microarray Data Analysis*, pages 69–80, 2002.

[3] K. Baggerly, J. Morris, and K. Coombes. Reproducibility of SELDI-TOF protein patterns in serum: comparing datasets from different experiments. *Bioinformatics*, 20(5):777–785, 2004.

[4] A. Bamgbade, R. Somorjai, B. Dolenko, E. Pranckeviciene, A. Nikulin, and R. Baumgartner. Evidence accumulation to identify discriminatory signatures in biomedical spectra. In S. Miksch, J. Hunter, and E. Keravnou, editors, *Proceedings of the Tenth Conference on Artificial Intelligence in Medicine (AIME 2005)*, pages 463–467, 2005.

[5] M. Beibel. Selection of informative genes in gene expression based diagnosis: A nonparametric approach. In R. Brause and E. Hanisch, editors, *Proceedings of the First International Symposium in Medical Data Analysis*, volume LNCS 1933, pages 300–307. Springer-Verlag, 2002.

[6] R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

[7] B. Bozkurt and D. Mann. Use biomarkers in the management of heart failure: Are we there yet? *Circulation*, 107:1231–1233, 2003.

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

[9] L. Breiman. Randomizing output to increase prediction accuracy. *Machine Learning*, 40(3):229–242, 2000.

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

[11] C. Chang and C. Lin. Libsvm: a library for support vector machines, 2001.

[12] T. Dietterich. Machine learning research: four current directions. *AI Magazine*, 18(4):97–136, 1997.

[13] T. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *Multiple Classifier Systems*, volume LNCS 1857, pages 1–15. Springer-Verlag, 2000.

[14] C. Ding. Analysis of gene expression profiles: class discovery and leaf ordering. In *Proceedings of the Sixth International Conference on Research in Compuational Molecular Biology*, pages 127–136, April 2002.

[15] R. Duda, P. hart, and D. Stork. *Pattern Classification*. Wiley-Interscience, second edition, 2000.

[16] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, Inc., 1993.

[17] A. Fred and A. K. Jain. Data clustering using evidence accumulation. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pages 545–554, August 2002.

[18] A. Fred and A. K. Jain. Combining multiple clusterings using evidence accumulation. *IEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, June 2005.

[19] Y. Freund. Boosting a weak learning algorithm by majority. In *Proceedings of the Third Annual Workshop on Computational Learning Theory*, pages 202–216, 1990.

[20] N. Schraudolph G. Orr and F. Cummins. Overfitting, 1999.

[21] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[22] S. Gunter. Feature selection algorithms for the generation of multiple classifier systems and their application to handwritten word recognition. *Pattern Recognition Letters*, 25(11):1323–1336, 2004.

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

[24] I. Guyon, J. Weston, and S. Barnhill. Gene selection for classification using support vector machines. *Machine Learning*, 46:389–422, 2002.

[25] T. Ho. The random subspace method for constructing decision forests. *IEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, August 1998.

[26] A. Jain, R. Duin, and J. Mao. Statistical pattern recognition: A review. *IEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, January 2000.

[27] A. Jain and D. Zongker. Feature selection: Evaluation, application, and small sample performance. *IEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):153–158, February 1997.

[28] A. K. Jain and B. Chandrasekaran. Dimensionality and sample size considerations in pattern recognition practice. In P. R. Krishnaiah and L. N. Kanal, editors, *Handbook of Statistics*, pages 224–234, 1982.

[29] N. Kasabov. *Evolving Connectionist Systems: Methods and applications in Bioinformatics, Brain Study and Intelligent Machines*. Springer-Verlag New York, Inc., 2002.

[30] R. Kohavi and G. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97:273–324, 1997.

[31] I. Levner. Feature selection and nearest centroid classsification for protein mass spectrometry. *BMC Bioinformatics*, 6(1), 2005.

[32] J. Li and K. Ramamohanarao. A tree-based approach to the discovery of diagnostic biomarkers for ovarian cancer. In H. Dai, R. Srikant, and C. Zhang, editors, *Advances in Knowledge Discovery and Data Mining, 8th Pacific-Asia Conference, Pacific-Asia Conference on Knowledge Discovery and Data Mining PAKDD 2004*, volume 3056 of *Lecture Notes in Computer Science*, pages 682–691, 2004.

[33] L. Li, C. Weinberg, T. Darden, and L. Pedersen. Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics*, 17(12):1131–1142, 2001.

[34] A. Nikulin, B. Dolenko, T. Bezabeh, and R. Somorjai. Near-optimal region selection for feature space reduction: Novel preprocessing methods for classifying mr spectra. *NMR in Biomedicine*, 11(4-5):209–216, 1998.

[35] L. Oliveira, R. Sabourin, F. Bortolozzi, and C. Suen. Feature selection for ensembles: a hierarchical multi-objective genetic algorithm approach. In *Proceedings of the Seventh International Conference on Document analysis and Recognition*, pages 676–680, 2003.

[36] D. Opitz. Feature selection for ensembles. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 379–384, 1999.

[37] E. Pranckeviciene, R. Somorjai, R. Baumgartner, and M. Jeon. Identification of signatures in biomedical spectra using domain knowledge. *Artificial Intelligence in Medicine (In press)*, 13(3):252–264, 2005.

[38] P. Pudil, J. Novovicova, and J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119–1125, 1994.

[39] M. Restelli, D. Sorrenti, and F. Marchese. Evidence accumulation method for mobile robot localization. In *Proceedings of the 8th conference of the Associazione Italiana per l'Intelligenza Artificiale*, 2002.

[40] W. Siedlecki and J. Sklansky. On automatic feature selection. In C.H Chen, L.F Oay, and P.S.P Wang, editors, *Handbook of Pattern Recognition and Computer Vision*, pages 63–88. World Scientific, 1993.

[41] R. Simon. Diagnostic and prognostic prediction using gene expression profiles in high-dimensional microarray data. *British Journal of Cancer*, 89(9):1599–1604, 2003.

[42] R. Simon. Pitfalls in the use of DNA microarray data for diagnostics and prognostic classification. *Journal of the National Cancer Institute*, 95(1):14–18, January 2003.

[43] R. Simon. Supervised analysis when the number of candidate features (p) greatly exceeds the number of cases (n). *Special Interest Group on Knowledge Didcovery in Data SIGKDD Explorations Newsletter*, 5(2):31–36, 2003.

[44] R. Simon. When is a genomic classifier ready for prime time? *Nature Clinical Practice – Oncology*, 1(1):4–5, November 2004.

[45] R. Simon and D. Altman. Statistical aspects of prognostic factor studies in oncology. *British Journal of Cancer*, 69(6):979–985, 1994.

[46] M. Skurichina. *Stabilizing Weak classifiers : regularization and combining techniques in Discriminant Analysis*. PhD thesis, Department of Data Analysis, Institute of Mathematics and Informatics, Vilnius,Lithuania, 2001.

[47] R. Somorjai, M. Alexander, R. Baumgartner, S. Booth, S. Bowman, C. Demko, B. Dolenko, M. Mandelzweig, A. Nikulin, N. Pizzi, E. Pranckeviciene, R. Summers, and P. Zhilkin. A data-driven, flexible machine learning strategy for the classification of biomedical data. In F. Azuaje and W. Dubitzky, editors, *Artificial Intelligence Methods and Tools for Systems Biology*, pages 67–85, 2004.

[48] R. Somorjai, B. Dolenko, and R. Baumgartner. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: Curses, caveats, cautions. *Bioinformatics*, 19(12):1484–1491, 2003.

[49] R. Somorjai and A. Nikulin. The curse of small sample sizes in medical diagnosis via MR spectroscopy. In *Proceedings of the Twelfth Annual Scientific Meeting of the Society for Magnetic Resonance in Medicine*, page 685, 1993.

[50] P. Srinivas, B. Kramer, and S. Srivastava. Trends in biomarker research for cancer detection. *The Lancet Oncology*, 2:698–704, 2001.

[51] P. Srinivas, M. Verma, Y. Zhao, and S. Srivastava. Proteomics for cancer biomarker discovery. *Clinical Chemistry*, 48:1160–1169, 2002.

[52] I. Iñza, B. Sierra, R. Blanco, and P. Larrañaga. Gene selection by sequential wrapper approaches in microarray cancer class prediction. *Journal of Intelligent and Fuzzy Systems*, 12(1):25–34, 2002.

[53] I. Iñza, P. Larrañaga, R. Blanco, and A. Cerrolaza. Filter versus wrapper gene selection approaches in DNA microarray domains. *Journal of Artificial Intelligence in Medicine*, 31:91–103, 2004.

[54] A. Tsymbal, P. cunningham, M. Pechenizkiy, and S. Puuronen. Search strategies for ensemble feature selection in medical diagnostics. In *Proceedings of the Sixteenth IEEE symposium on Computer-Based Medical Systems*, pages 124–129, 2003.

[55] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishiad, R. Spang, H. Zuzan, J. Olsonjr, J. Marks, and J. Nevins. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences in the United States of America*, 98(20):11462–11467, 2001.

[56] D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[57] E. Xing, M. Jordan, and R. Karp. Feature selection for high-dimensional genomic microarray data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 601–608, 2001.

[58] M. Xiong, X. Fang, and J. Zhao. Biomarker identification by feature wrappers. *Genome Research*, 11:1878–1887, 2001.

[59] M. Xiong, W. Li, J. Zhao, L. Jin, and E. Boerwinkle. Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism*, 73:239–247, 2001.