Memetic algorithms for feature/gene selection

Zhu, Zexuan

2007

Zhu, Z. X. (2007). Memetic algorithms for feature/gene selection. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/2489>

<https://doi.org/10.32657/10356/2489>

Nanyang Technological University

Downloaded on 26 Apr 2023 20:32:23 SGT

Memetic Algorithms for Feature/Gene Selection

Zhu Zexuan

School of Computer Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2007

Acknowledgements

First and foremost I would like to express my sincere gratitude and recognition to my supervisor, Dr. Ong Yew Soon, for his utmost patience, invaluable instructions and encouragement in the past few years. Without his support, my research would not reach this stage.

I am grateful to Dr. Manoranjan Dash, Dr. Ng See Kiong (I2R, A-Star Research Center) and Dr. Wang Kok Wai for their precious guidance and suggestions on my research.

I would also like to thank all the members of Bioinformatics Research Center for their friendship and creating a comfortable research environment, especially to the technicians, thanks for their unhesitating and professional help for addressing the technical problems during my research.

I would extend my appreciation to Ms. Halimabee binte Jainul Hussain and Ms. Cao Li for their collaborations in the early year of my candidature.

Thanks to my friends for their consistent supports and kindly help. A special thank to Dr. Ma Jianmin, Dr. Duan Kaibo, Cao Runzi, Yang Xiao, Zheng Yun, Zhu Ning, Zhou Zongzhao, and Ma Li for every encouragement and insightful discussion.

Thank all of them for sharing with me many joyful memory and experiences.

I am indebted to my girl friend Cai Te for her selfless love. I had you accompany with me whenever I felt frustrated or sad. I would never forget the happy time we shared with each other in Singapore.

Last but not least, I could never thank my family enough for their endless love and warm caring of my life. Thanks for their unconditional supports and encouragement whenever I face the difficulties or challenges during my study in the graduate school.

List of Abbreviations

CV	Cross-Validation
EA	Evolutionary Algorithm
FCR	Full Class Relevance
GA	Genetic Algorithm
KNN	K-Nearest Neighbor
LOOCV	Leave-One-Out Cross-Validation
LS	Local Search
MA	Memetic Algorithm
MBEGA	Markov Blanket Embedded Genetic Algorithm
MBE-MOMA	Markov Blanket Embedded Multiobjective Memetic Algorithm
MOMA	Multiobjective Memetic Algorithm
MO	Multiobjective Optimization
MOEA	Multiobjective Evolutionary Algorithm
MOP	Multiobjective Optimization Problem
OVA	One-Versus-All
PCR	Partial Class Relevance

SABMA	Sensitivity Analysis Based Memetic Algorithm
SVM	Support Vector Machine
SVS	Subset-Versus-Subset
WFFSA	Wrapper-Filter Feature Selection Algorithm

Table of contents

Acknowledgements	i
List of Abbreviations	iii
List of Figures	x
List of Tables	xiii
Abstract	xvi
1 Introduction	1
1.1 Problems and Motivations	2
1.2 Organization of this Dissertation	6
2 Background	8
2.1 Feature Selection	8
2.1.1 Feature Relevance and Redundancy	9
2.1.2 Filter vs. Wrapper Method	12
2.1.3 Search the Feature Space	14

2.2	Genetic Algorithm	15
2.3	Local Search	17
2.4	Memetic Algorithm	18
2.5	Evolutionary Multiobjective Optimization Algorithm	22
2.6	Generalization Error Estimation and Overfitting	26
2.6.1	k -Fold Cross-Validation	26
2.6.2	Bootstrap	27
2.6.3	Overfitting	27
2.6.4	Internal and External Validation	28
2.7	Microarray	30
2.7.1	Microarray Technology	31
2.7.2	Challenges of Analyzing Microarray Data	32
3	Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework	34
3.1	Introduction	35
3.2	Wrapper-Filter Feature Selection Algorithm	36
3.2.1	Encoding Representation and Initialization	36
3.2.2	Objective Function	38
3.2.3	Local Search Improvement Procedure	39
3.2.4	Evolutionary Operators	42
3.2.5	Computational Complexity	44
3.3	Experimental Results and Discussions	46

3.3.1	Comparison of Filter, GA, and WFFSA	48
3.3.2	Study on LS Strategies	52
3.4	Summary	55
4	Markov Blanket-Embedded Genetic Algorithm for Gene Selection	61
4.1	Introduction	62
4.2	System and Methodology	64
4.2.1	Markov Blanket	64
4.2.2	Markov Blanket Embedded GA	66
4.3	Empirical Study	70
4.3.1	Empirical Results on Synthetic Data	72
4.3.2	Experimental Results on Microarray Data	75
4.4	Summary	89
5	Comparing Memetic Algorithms For Gene Selection	91
5.1	Sensitivity Analysis Based MA	92
5.2	Comparison Study on WFFSA, MBEGA, and SABMA	95
5.2.1	Synthetic Data	96
5.2.2	Microarray Data	98
5.2.3	Robustness	100
5.3	Summary	100
6	Simultaneous Identification of Full Class Relevant and Partial Class	
	Relevant Genes	103

6.1	Introduction and Motivation	103
6.2	Multiclass Relevant Features	106
6.2.1	Multiclass Relevant Features	107
6.2.2	Full/Partial Class Relevant	107
6.3	System and Methodology	110
6.3.1	Problem Formulation	110
6.3.2	Markov Blanket Embedded Multiobjective Memetic Algorithm	111
6.3.3	Synergy Between FCR and PCR Feature Subsets	115
6.4	Empirical Study	116
6.4.1	Synthetic Data	117
6.4.2	Microarray Data	124
6.5	Summary	134
7	Conclusions	136
7.1	Contributions	136
7.2	Future Work	138
	Author's Publications	141
	Appendices	143
A	Filter Feature Selection Methods	143
A.1	ReliefF Feature Selection	143
A.2	Gain Ratio Feature Selection	144
A.3	Chi Square Feature Selection	145

B	Classification Methods	147
B.1	K-nearest Neighbor Classification	147
B.2	Support Vector Machine	148
B.2.1	Radius Margin Bound	149
C	An Example of FCR and PCR Features	151
D	Microarray Datasets	153
	Bibliography	154

List of Figures

2.1	Feature selection procedure	9
2.2	Filter feature selection	12
2.3	Wrapper feature selection	13
2.4	Outline of a genetic algorithm	15
2.5	Local search	17
2.6	Outlines of a memetic algorithm	20
2.7	Outline of multiobjective evolutionary algorithm	25
2.8	Overview of microarray technology	31
3.1	The Procedure of WFFSA	37
3.2	Representation of a chromosome as a binary string.	37
3.3	Add and Del operations (X and Y denote the selected and excluded feature subsets, respectively).	40
3.4	Procedure of improvement first strategy.	42
3.5	Procedure of steepest descent strategy.	43
3.6	Procedure of restrictive crossover.	44
3.7	An example of alleles to undergo restrictive crossover operation. . . .	44

3.8	Procedure of restrictive mutation.	45
3.9	Search trace on Vehicle dataset.	56
3.10	Search trace on WDBC dataset.	56
3.11	Search trace on Ionosphere dataset.	57
3.12	Search trace on Sonar dataset.	57
3.13	Search trace on ALL-AML1000 dataset.	58
3.14	Search trace on Colon1000 dataset.	58
3.15	Search trace on NCI1000 dataset.	59
3.16	Search trace on SRBCT dataset.	59
4.1	Markov blanket embedded genetic algorithm (MBEGA) for gene selection	66
4.2	<i>Add</i> operation	68
4.3	<i>Del</i> operation	68
4.4	Markov blanket based memetic operations (AMB denotes Approximate Markov Blanket)	69
4.5	The Z-score of the top 50 most frequently selected genes (a) Colon Cancer, (b) Central Nervous System, (c) ALL-AML, (d) Breast Cancer, (e) Lung Cancer, (f) Ovarian Cancer.	85
4.6	The Z-score of the top 50 most frequently selected genes (a) ALL-AML-3, (b) ALL-AML-4, (c) Lymphoma, (d) MLL, (e) SRBCT. . . .	86
5.1	<i>Add</i> operator for SABMA	94
5.2	<i>Del</i> operator in SABMA	95

5.3	The Z-score of the top 50 most frequently selected genes on (a) Colon, (b) Central Nervous System, (c) ALL-AML, (d) Breast, (e) Lung, (f) Ovarian, (g) Synthetic.	101
6.1	Outline of Markov blanket embedded Multiobjective Memetic Algo- rithm for Feature Selection	112
6.2	Markov blanket based LS in MBE-MOMA	114
6.3	The three-dimension Pareto front on the SD2	123
6.4	Two-dimension projections of the Pareto front. (a) Objective 1 vs. Objective 2, (b) Objective 1 vs. Objective 3, (c) Objective 2 vs. Objective 3. $Gx \ll Gy$ ($x, y = 1, 2, 3$) indicates more genes are selected from group y than x . $Gx \approx Gy$ ($x, y = 1, 2, 3$) indicates similar number of genes are selected from group x and y ,	125
A.1	Outlines of ReliefF	144

List of Tables

3.1	Datasets and Parameters Used for Experiments	48
3.2	Comparison Results of Filter, GA, and WFFSA Methods	51
3.3	Results by WFFSAs and Other Methods on UCI Datasets	52
3.4	Results by WFFSAs and Other Methods on Microarray Datasets . . .	52
3.5	Results of Different LS Strategies, Length and Interval on Sonar Dataset	53
3.6	Results of Different LS Strategies, Length and Interval on Colon1000 Dataset	53
3.7	Results of Different LS Strategies on All the Eight Datasets	55
4.1	Summary of the four synthetic datasets	74
4.2	Feature selection by each algorithm on synthetic data	74
4.3	Description of 11 Microarray Datasets	76
4.4	Performance of feature selection algorithms on microarray datasets .	79
4.5	Correlation coefficient between running time and other external factors	81
4.6	Summation ranks of four evaluation criteria	87
5.1	Feature selection by each algorithm on synthetic data	97

5.2	Average of classification accuracy and selected gene sized of feature selection algorithms, GA, WFFSA, SAMMA, and MBEGA, on microarray datasets	99
6.1	Classification accuracy by each algorithm on synthetic data	121
6.2	Feature selected by each algorithm on synthetic data	122
6.3	Description of 10 Multiclass Microarray Datasets	126
6.4	Classification accuracy of feature selection algorithms on ten multiclass microarray datasets	126
6.5	20 most selected FCR genes on the ALL-AML-3 dataset	128
6.6	10 most selected PCR genes relevant to (ALL B-cell vs. Others) on the ALL-AML-3 dataset	129
6.7	10 most selected PCR genes relevant to (ALL T-Cell vs. Others) on the ALL-AML-3 dataset	130
6.8	10 most selected PCR genes relevant to (AML vs. Others) on the ALL-AML-3 dataset	130
6.9	5 selected FCR genes on the Thyroid dataset	131
6.10	10 most selected PCR genes relevant to (Follicular adenoma vs. Others) on the Thyroid dataset	131
6.11	10 most selected PCR genes relevant to (Follicular carcinoma vs. Others) on the Thyroid dataset	132
6.12	10 most selected PCR genes relevant to (Papillary Carcinoma vs. Others) on the Thyroid dataset	132

6.13 10 most selected PCR genes relevant to (Normal vs. Others) on the Thyroid dataset	133
C.1 A Simple Example of FCR and PCR Features	151
D.1 Description of ALL Microarray Datasets Used in This Dissertation . .	153

Abstract

This dissertation presents novel memetic frameworks for the hybridization of wrapper and filter feature selection methods on classification problems. The frameworks incorporate filter methods in the traditional genetic algorithm (GA) to improve classification performance and accelerate the search in identifying the core feature subsets. Particularly, the filter methods are introduced to add or delete features from a candidate feature subset encoded in a GA solution. Using memetic frameworks, we propose three feature selection algorithms, Wrapper-Filter Feature Selection Algorithm (WFFSA), Markov Blanket Embedded Genetic Algorithm (MBEGA), and Markov Blanket Embedded Multiobjective Memetic Algorithm (MBE-MOMA).

WFFSA fine-tunes the population of GA solutions by adding and deleting features based on a univariate feature filter ranking method (e.g., ReliefF, Gain Ratio, and Chi-square). Empirical studies on synthetic and real-world (UCI and microarray) datasets suggest WFFSA identifies the relevant features more efficiently than many existing methods in the literature for mining problems with a large number of features. At the same time, it converges to improved or competitive classification accuracy with smaller number of identified features.

MBEGA embeds the Markov blanket to fine-tune the search by adding relevant features and removing redundant and/or irrelevant features in the GA solutions. In contrast to WFFSA, it is targeted to handle problems with large number of irrelevant and redundant features more effectively and efficiently. Studies on MBEGA indicate that it attains better accuracy and robustness than other feature selection algorithms on microarray datasets since such data contains very large number of redundant features.

Finally, MBE-MOMA represents a synergy between the multiobjective evolutionary algorithm and Markov blanket-based local search. It is designed for the simultaneous identification of both full class relevant (FCR) and partial class relevant (PCR) features in multiclass problems. MBE-MOMA is demonstrated to be significantly more efficient in the identification of both FCR and PCR features on multiclass microarray datasets. In particular, in terms of classification accuracy, it scores better or competitive to the single objective MBEGA counterpart, MOEA without the Markov blanket-based local search, as well as some existing methods in the literature, especially on datasets with a lower percentage of FCR features.

Chapter 1

Introduction

Feature selection has become the focus of many research areas in recent years. With the rapid advance of computer and database technologies, datasets with hundreds and thousands of variables or features are now ubiquitous in pattern recognition, data mining, and machine learning [1, 2]. To process such huge datasets is a challenging task because traditional machine learning techniques usually work well only on small datasets. Feature selection addresses this problem by removing the irrelevant, redundant, or noisy features. It improves the performance of the learning algorithms, reduces the computational cost, and provides better understandings of the datasets.

Feature selection on microarray data has attracted increasing interest in many academic communities and industries over the last decade. The breakthrough in microarray technology promises a new insight into the mechanisms of living systems by providing a way to simultaneously measure the activities and interaction of thousands of genes. The main difficulties of analyzing microarray data lie in its inherently noisy and high-dimensional nature. Microarray data is characterized with thousands of genes but with only a small number of samples available for analysis.

This makes learning from microarray data an arduous task under the effect of curse of dimensionality. Furthermore, microarray data often contains many irrelevant and redundant features, which affect the speed and accuracy of most learning algorithms. Hence, feature selection is widely used to addresses these problems in microarray data analysis, where feature selection is also known as gene selection.

In this dissertation, we propose novel feature selection methods for gene selection using the concept of memetic algorithm (MA) [3–7]. In particular, these new feature selection methods are synergies of filter and GA wrapper methods using a memetic framework. Taking advantage of both filter and wrapper methods, they search the feature space much more efficiently and converges to small robust feature subsets with high prediction accuracy.

1.1 Problems and Motivations

Feature selection is defined as the problem of selecting a minimal subset of M features from the original set of N features ($M \leq N$), so that the feature space is optimally reduced and the performance of the learning algorithm is improved or not significantly decreased [1,8–10]. Feature selection algorithms are widely categorized into two groups, i.e., 1) filter methods and 2) wrapper methods [8,11]. Filter methods evaluate the goodness of a feature subset by using the intrinsic characteristic of the data. They are relatively computationally cheap since they do not involve the induction algorithm. However, they also take the risk of selecting subsets of features that may not match the chosen induction algorithm. Wrapper methods, on the

contrary, directly use the induction algorithm to evaluate feature subsets. They generally outperform filter methods in terms of prediction accuracy, but the trade-off is a higher computational effort.

Another key issue for feature selection algorithms is how to search the space of feature subsets which is exponential in the number of features. The complexity of searching the optimal feature subset is $O(2^N)$ and many problems related to feature selection are NP-hard problems. Many traditional search methods have shown promising results in a number of feature selection problems. However, as the number of features increases extremely (in this thesis we focus on high-dimensional microarray data), most of these existing methods face the problem of intractable computational time. On the other hand, most of these features in the dataset are often found to be irrelevant and redundant. Keeping such features would inevitably deteriorate the performance of induction algorithms.

In summary, in order to apply feature selection effectively and efficiently to high-dimensional datasets, one generally has to address the following issues:

1. Which schema should we use on a given problem? Filter or Wrapper?
2. How to search the feature space efficiently for producing a good prediction accuracy?
3. How to deal with irrelevant and redundant features?

In this thesis, we begin with an attempt to answer the first two questions. Among the wrapper methods, genetic algorithm (GA) [12] represents one of the commonly

used modern stochastic global search techniques for feature selection, due to its ease of implementation and its ability to converge close to the global optimal solution within a tractable time. It has shown promising performance in producing high quality solutions within tractable time on complex problems [13–21]. However, due to the inherent nature of GA, it often takes a long time to locate the local optimum in a region of convergence and sometimes may not find the optimum with sufficient precision. An increasing practice to improve the efficiency of the standard GA is to incorporate some memetic operators or local searches that are capable of fine-tuning and improving the solutions generated by the GA throughout the evolutionary search. This form of hybrid evolutionary algorithm is now commonly referred to as Memetic algorithm [3–7].

Since filter methods are generally cheaper to compute, they make very suitable memes or local searches for MA feature selection. Based on the aforementioned motivations, we propose a wrapper-filter feature selection algorithm (WFFSA) [22] using a memetic framework [3–7]. In particular, a filter ranking method based local search is introduced to fine-tune the population of GA solutions by adding or deleting features based on univariate feature ranking information. Empirical studies of WFFSA on several commonly used UCI [23] and microarray datasets indicate that it outperforms several recently reported feature selection methods in the literature in terms of classification accuracy, selected feature size, and search efficiency.

However, because the filter methods (i.e., ReliefF [24], Gain Ratio [25], and Chi-square [26]) used in WFFSA are forms of univariate ranking approaches, they

generally fail in identifying possible interactions between features and hence may not be capable of identifying redundant features efficiently. An alternative to univariate ranking methods is the use of Markov blanket [27]. Markov blanket is a cross-entropy based filter technique capable of identifying both redundant and irrelevant features [27, 28]. Using the cross-entropy based method, we propose an alternative MA feature selection method known here as the Markov blanket embedded GA (MBEGA) [29], which is a hybridization of the Markov blanket with GA wrapper methods. The empirical studies on microarray data suggest that MBEGA is more effective and efficient in eliminating irrelevant and redundant genes than several other gene selection algorithms including WFFSA. A detailed comparative study with other methods from each of filter, wrapper, and standard GA shows that MBEGA gives a best compromise among all four evaluation criteria, i.e., classification accuracy, number of selected genes, computational cost, and robustness.

On the other hand, multiclass problems on microarray data have drawn increasing attention in recent years. Multiclass cancer classification has provided the feasibility of cancer diagnosis across all of the common malignancies in parallel. Using multiclass cancer feature selection approaches, it is now possible to identify genes that are relevant to a set of cancer types. However, besides identifying the relevant genes, it is also useful for biologists to know which specific cancer or subset of cancer types a gene is relevant to. In this paper, we introduce two new definitions of multiclass relevancy features, i.e., full class relevant (FCR) and partial class relevant (PCR) features. Particularly, FCR denotes genes that serve as candidate biomark-

ers for discriminating any cancer types. PCR, on the other hand, represents genes that distinguish subsets of cancer types. It is worth noting that the use of both FCR and PCR genes can help improve the classification accuracy of the machine learner hence providing accurate prediction on cancer diagnosis. However, little efforts in the field on multiclass gene selection have demonstrated the importance and identification of PCR genes. Most of the state-of-art multiclass gene selection algorithms, including MBEGA, were not designed to identify PCR genes. In this dissertation, we propose a Markov blanket embedded multiobjective memetic algorithm (MBE-MOMA) [30] for simultaneous identification of both FCR and PCR genes in multiclass microarray cancer data. In particular, MBE-MOMA optimizes multiple objectives simultaneously with each objective corresponding to the classification accuracy for feature/gene selection on an one-versus-all problem. This new method is shown to be efficient for identifying both FCR and PCR genes that may aid biologists in their study at the same time improving the classification accuracy of the induction algorithm.

1.2 Organization of this Dissertation

The remainder of this dissertation is organized as six chapters.

Chapter 2 provides an overview on the basic concepts of feature selection, genetic algorithm, local search, memetic algorithm, evolutionary multiobjective optimization algorithm, performance validation schemes commonly used in machine learning, and microarray technology. Chapter 3 describes the proposed WFFSA based on a

memetic framework. Chapter 4 describes our proposed algorithm MBEGA. Chapter 5 compares the performance of three MA feature selection methods, i.e., WFFSA, MBEGA and Sensitivity Analysis Based MA (SABMA), on microarray data. Chapter 6 introduces the notions of FCR and PCR features as well as the proposed MBE-MOMA feature selection method for efficient simultaneous identification of FCR and PCR features on multiclass problems. Finally, Chapter 7 concludes this thesis.

Chapter 2

Background

In this chapter, we provide some background knowledge on feature selection, genetic algorithms, local search, memetic algorithm, evolutionary multiobjective optimization algorithm, generalization error estimation, and microarray technology.

2.1 Feature Selection

Feature selection involves the problem of selecting a minimal subset of M features from the original set of N features ($M \leq N$), so that the feature space is optimally reduced and the performance of the learning algorithm is improved or not significantly decreased [1, 8–10]. Generally, a typical feature selection method consists of four components: a generation procedure or search procedure, evaluation function, stopping criterion, and validation procedure. Every time a candidate subset of features is generated randomly or based on some heuristics, it is evaluated based on some independent (i.e., without involving any induction algorithm) or dependent criteria (i.e., performance of the induction algorithm). This procedure of feature selection is repeated until a predefined stopping condition is satisfied, which may be due to the completion of search, an achievement of a sufficiently good subset,

or violation of the maximum allowable number of iterations. Next, the selected subset should be validated either using prior knowledge about the data or using some unseen independent testing dataset. This general process of feature selection is illustrated in Figure 2.1. In this dissertation, we focus on feature selection for classification. Therefore, the induction algorithm in this study is usually a classifier, e.g., K-Nearest Neighbors (KNN), Decision Trees, or Support Vector Machine (SVM).

Figure 2.1: Feature selection procedure

2.1.1 Feature Relevance and Redundancy

The goal of feature selection is to select an optimal feature subset. Let X be a full set of features, X_i be a feature, and $\overline{X}_i$ be the feature set of all features except X_i , i.e., $\overline{X}_i = X - \{X_i\}$. An instance is typically described as an assignment of values $x = (x_1, \dots, x_N)$ to a set of features $X = (X_1, \dots, X_N)$ and a class label Y , which takes one of the k possible values in class set $C = \{c_1, \dots, c_k\}$. A value-assignment

to all features in $\overline{X}_i$ is denoted by $\overline{x}_i$. Here, the optimal feature subset as defined in Yu and Liu [2, 28] is given as follows:

Definition 2.1 *Optimal Subset:* Let G be a subset of X , x_G and x be the value vector of G and X , respectively. G is called an optimal subset if and only if G is a minimum subset such that $P(Y = C|G = x_G)$ is equal or as close as possible to $P(Y = C|X = x)$.

where $P(Y = C|G = x_G)$ is the probability distribution of different classes given the feature values in G and $P(Y = C|X = x)$ is the original distribution given the feature values in X .

The relevance and redundancy of features were first defined in [11] and [27], respectively. Kohavi et al. [11] classified features into three disjoint categories, namely, strongly relevant, weakly relevant, and irrelevant features.

Definition 2.2 *Strong Relevance:* A feature X_i is strongly relevant if and only if there exists some x_i , c_j , and $\overline{x}_i$ for which $p(X_i = x_i, \overline{X}_i = \overline{x}_i) > 0$ such that $p(Y = c_j|X_i = x_i, \overline{X}_i = \overline{x}_i) \neq p(Y = c_j|\overline{X}_i = \overline{x}_i)$.

Definition 2.3 *Weak Relevance:* A feature X_i is weakly relevant if and only if it is not strongly relevant and there exists a subset of features $\overline{X}'_i \subseteq \overline{X}_i$ for which there exists some $\overline{x}_i$, c_j , and $\overline{x}'_i$ for which $p(X_i = x_i, \overline{X}'_i = \overline{x}'_i) > 0$ such that $p(Y = c_j|X_i = x_i, \overline{X}'_i = \overline{x}'_i) \neq p(Y = c_j|\overline{X}'_i = \overline{x}'_i)$.

Therefore, a feature is **relevant** if it is either strongly relevant or weakly relevant, otherwise, it is **irrelevant**. In practice, an optimal subset essentially contains all the

strongly relevant features and some weakly relevant features that are not redundant. However, based on the definition in [11], it remains unclear which are redundant among the weakly relevant features. Thus, it is necessary to further define the redundancy among relevant features. To introduce the concept of redundant feature, we begin with a brief overview of Markov blanket [27, 31].

Definition 2.4 Markov Blanket: Let M be a subset of features which does not contain X_i , i.e., $M \subseteq X$ and $X_i \notin M$. M is a Markov blanket of X_i if X_i is conditionally independent of $(X \cup C) - M - \{X_i\}$ given M , i.e., $P(X - M - \{X_i\}, C | X_i, M) = P(X - M - \{X_i\}, C | M)$.

Definition 2.5 Redundant feature: Let G be the current set of selected features, a feature F_i is redundant and hence should be removed from G if and only if it is weakly relevant and has a Markov blanket M in G .

Currently, feature selection research has focused on dealing with redundant features and searching for the optimal feature subset [2]. Koller and Sahami [27] proposed a correlation based feature selection method and Markov blanket for identifying and eliminating redundant features. However, the computational complexity for finding the conditional independence of features in Markov blanket is typically very high, which limits the efficiency of the method. To mitigate this deficiency of Markov blanket, Yu and Liu [28] considered using only one feature to approximate the Markov blanket by proposing the efficient fast correlation-based filter method (FCBF). FCBF was demonstrated as being superior to other popular feature selec-

tion methods including ReliefF [24], CFS-SF (variation of CFS [32]) and FOCUS-SF (variation of FOCUS [33]).

2.1.2 Filter vs. Wrapper Method

In general, feature selection algorithms are widely categorized into two groups: filter and wrapper methods [11].

Figure 2.2: Filter feature selection

Filter methods evaluate the relevance or goodness of the feature subset by using the intrinsic characteristic of the training data, ignoring the induction algorithm (as shown in Figure 2.2). They usually evaluate each feature based on simple statistics obtained from the training data distribution (e.g., Chi-Square [26], Information Gain [34], Gain Ratio [25], ReliefF [24], etc.). Filter methods are relatively computationally cheap, since they do not involve any induction algorithm; however, they are subjected to the risk of selecting subsets of features which may not match the chosen induction algorithm.

Wrapper methods, on the contrary, directly use the induction algorithm to evaluate the feature subsets (as shown in Figure 2.3). The feature subset selection algorithm conducts a search for a good subset using the induction algorithm as part

Figure 2.3: Wrapper feature selection

of the evaluation function. Hence, wrapper methods generally select a more suitable feature subset for the induction algorithm than filter methods and also outperform in terms of prediction accuracy. The trade-off however is that wrapper methods are generally more computationally expensive.

In effect, some researchers have tried to combine the advantages of both filter and wrapper methods using hybrid models. For example, a simple integration of a filter method as a pre-selection step and a wrapper method for feature selection was recently introduced in the ORDERED-FS algorithm [35]. Nevertheless, ORDERED-FS may introduce bias and miss out of some important features of the induction algorithm in the filter pre-processing stage, leading to possible poor performance. In [36], Das proposed another hybrid feature selection method, BBHFS, using a boosting algorithm. In particular, BBHFS is built on a filter method, information gain, and the induction algorithm, decision stumps, is introduced to provide a stopping criterion as well as to assign weights for samples during boosting. In this dissertation, we proposed several novel hybrid filter and wrapper algorithms for handling the classification problems using memetic algorithms (MAs). The proposed algorithms are combinations of genetic algorithm (GA) and filter-based local search.

In particular, the filter-based local search method fine-tunes the population of GA wrapper method by adding or deleting features on the solutions. Greater details on GA and MA are later described in Sections 2.2 and 2.4.

2.1.3 Search the Feature Space

The key issue for feature selection methods, especially for wrapper methods, is how to search on feature space. Ideally, a feature selection method searches through the subsets of features and tries to find the best among the competing 2^N candidate subsets according to some evaluation criteria.

A number of search strategies have been proposed in the literature for feature selection. They can be categorized into complete search (e.g., FOCUS [33], Branch and Bound [37]), heuristic search (e.g., best-first method used in CFS [32], sequential search [38–40]), and random search (e.g., Random Search used in LVF [41], Las Vegas algorithm [42]). For a survey on different search methods used in feature selection, the reader is referred to [1, 2, 8, 10]. These methods have shown promising results in a number of applications. However, as the number of features N increases, most of these existing methods face the problems of intractable computational time and convergence to suboptimal feature subsets. Recently, an increasing number of nature inspired approaches such as genetic algorithms are used in feature selection due to their well-known ability to produce high quality solutions within a tractable time on complex search problems.

2.2 Genetic Algorithm

Genetic Algorithm

BEGIN

1. $t = 0$;
2. **Initialize:** Randomly generate an initial population $P(t)$;
3. Evaluate fitness of each solution in $P(t)$;
4. **While**(*Termination Criterion Not Fulfilled*)
 5. Select a temporary population $P'(t)$ for reproduction from population $P(t)$;
 6. Perform crossover and mutation on $P'(t)$;
 7. Evaluate fitness of each solution in $P'(t)$;
 8. Select solutions from $\{P(t) \cup P'(t)\}$ to create a new population $P(t+1)$;
 9. $t = t+1$;
10. **End While**

END

Figure 2.4: Outline of a genetic algorithm

Genetic Algorithm (GA) [12] has emerged as a powerful paradigm for global optimization. Over the last several decades, GA has been successfully applied to a wide class of complex search problems [13–21, 43–45]. Its popularity lies in ease of implementation and its ability to converge close to the global optimal solution within a tractable time.

Inspired by the power of natural evolution, GA conducts global search by maintaining a population of candidate solutions and evolving the population by subjecting it to replication, variation, and selection. The general template of an GA is outlined in Figure 2.4. In the first step, an initial population is created randomly

or using some design of experiment methods, so that it is uniformly distributed throughout the search space. Succeeding generations are produced from the previous by means of selection, crossover, and mutation.

GA has been widely used for feature selection [13,14,16–18,20,46,47]. Typically, in GA feature selection, each solution (or chromosome) in the population encodes a corresponding selected feature subset. One often uses binary encoding with a bit string of length equal to the total number of features [13,14,20] or integer encoding with an integer string of length equal to the number of selected features. In binary encoding, a bit of '1' ('0') implies the corresponding feature is selected (excluded). While, in integer encoding, each integer in the chromosome indicates the index of the selected feature among the whole feature set. The fitness of each chromosome c is then obtained using an objective function based on the induction algorithm, i.e.,

$$Fitness(c) = w_1 J(S_c) + w_2 |S_c| \quad (2.1)$$

where S_c denotes the selected feature subset encoded in the chromosome c , $J(S_c)$ evaluates the generalization error of the induction algorithm using the given feature subset S_c , and $|S_c|$ denotes the number of features in S_c . The fitness function is the linear combination of $J(S_c)$ and $|S_c|$. After calculating the fitness of each chromosome, GA evolves the population using selection, crossover, and mutation.

In many existing work, GA has shown promising performance in feature selection [13,14,16–18,20,46,47]. However, as the feature size increases, GA could take a long time to locate the local optimum in a region of convergence and may sometimes not

find the optimum with sufficient precision due to its inherent nature. One way to solve this problem is to hybridize GA with some local search operations (also known as memetic operations) [3, 48] which are capable of fine-tuning and improving the solutions generated by GA to make them more accurate and efficient. This form of evolutionary algorithm is referred as Memetic algorithm (MA), which will be described in Section 2.4.

2.3 Local Search

Before introducing memetic algorithms, we present some preliminary knowledge on local search in this section. Local search (LS) is the process of searching the near neighborhood region of a solution and adopting a better solution if it exists. The procedure of a typical LS method is illustrate in Figure 2.5. Different LS methods can be distinguished by their choices of neighborhood structure and move decision.

Local Search BEGIN

1. Given a solution s .
2. **Repeat:**
3. Generate the neighborhood structure of s ;
4. Move to a neighbor s' , i.e., $s = s'$, if s' is better than s ;
5. **Until:** s has no neighbors better than itself;

END

Figure 2.5: Local search

LS methods assume a neighborhood structure, which define what moves are

legal. A neighborhood structure of a given solution on combinatorial optimization is defined as the whole set of solutions that can be reached by applying an elementary operator to this solution. For example, in feature selection problems, the *bit-flip* is an elementary operator for binary encoding strings, which encode corresponding selected feature subsets. Therefore, the neighborhood structure of a given selected feature subset S_c consists of those feature subsets that can be generated by flipping one bit on chromosome c , i.e., adding(deleting) a feature to(from) S_c .

The second decision for LS methods is the choice of move decision. In other words, LS methods to examine which neighbors among the neighborhood structure are better than the current solution, and to decide which one the search would move to. For instance, two typical decision rules are, steepest descent (i.e., best move) and first-improvement (i.e., first move). The steepest descent method always moves to the best neighbor. It requires the whole neighborhood structure to be examined at each step. The first-improvement method simply moves to the first better neighbor it visits. Hence, the algorithm may examine only a fraction of the neighborhood structure.

2.4 Memetic Algorithm

It is now widely accepted that two competing goals govern the design of global search methods, exploration and exploitation. The former is important to ensure global reliability, i.e., every part of the domain is searched sufficiently to provide a reliable estimate of the global optimum. Exploitation on the other hand is also important since

it concentrates the search effort around the best solutions found so far by searching their neighborhood structure to derive better solutions. Taking this cue, most recent global search algorithms are now designed to achieve these two goals using a combination of dedicated global and local searches. This form of search algorithms may be regarded as a marriage between population-based global search (e.g., GA) and the local improvement made by each of the individuals (e.g., LS). It has the potential of exploiting the complementary advantages of GAs (generality, robustness, global search efficiency), and problem-specific LS (exploiting application-specific problem structure, rapid convergence toward local minima). Such combinations of optimizers are commonly known as Memetic Algorithms (MAs) [3, 48].

MAs are population-based meta-heuristic search methods inspired by Darwinian principles of natural evolution and Dawkins' notion of a meme defined as a unit of cultural evolution that is capable of local refinements. Recent studies on MAs have revealed their success on a wide variety of real world problems. Particularly, they not only converge to high quality solutions, but also search more efficiently than their conventional counterparts [3–7, 44, 49–54]. In diverse contexts, MAs are also commonly known as hybrid Evolutionary Algorithms (EAs), Baldwinian EAs, Lamarckian EAs, cultural algorithms and genetic LS. A typical MA can be described as Figure 2.6.

In order to locate the global optimum of a search problem accurately and efficiently under a limited computational budget, a good balance between exploration and exploitation in the MA must be appropriately maintained throughout the opti-

Memetic Algorithm
BEGIN

1. $t = 0$;
2. **Initialize:** Randomly generate an initial population $P(t)$;
3. Evaluate the fitness of each solution in $P(t)$;
4. **While**(*Termination Criterion Not Fulfilled*)
 5. Apply local search on all or part of solutions in $P(t)$;
 6. Evaluate the fitness of each solution in $P(t)$;
 7. Select a temporary population $P'(t)$ for reproduction from $P(t)$;
 8. Perform recombination and mutation on $P'(t)$;
 9. Select solutions from $\{P(t) \cup P'(t)\}$ to create a new population $P(t + 1)$;
10. $t = t + 1$;
11. **End While**

END

 Figure 2.6: Outlines of a memetic algorithm

mization search process. Neither exploitation nor exploration should dominate for optimum MA search performance. Over the recent years, a number of researchers have addressed several issues relating to the design of MAs, which include: 1) How often should local learning be applied, i.e., local search frequency? Alternatively, what is the probability of applying local learning to an individual? 2) On which solution subset of the population should the local learning be applied? 3) How long should the local learning be run, i.e., local search intensity or local search computational time budget? 4) Which local learning procedure or local search or meme should be used?

Hart [54] investigated the effect of local search frequency on MA search performance. Upon which, he investigated various configurations of the local search frequency at different stages of the MA search. He also touched on the issue of selecting appropriate individuals among the EA population that should undergo local search. In his work, fitness-based and distribution-based strategies were studied for adapting the probability of applying local search on the population of chromosomes in continuous parametric search problems. Hart made a conclusion that just some solutions are going to be used for LS, i.e., applying LS to every individual of the population is not necessary, since LS can be wasted on the points far away from a global optimum. Land [49] extended Hart's work to combinatorial optimization problems and introduced the concept of "sniff" for balancing genetic and local search or also known as the local/global ratio. Ishibuchi [6] and his colleagues extended the study to multiobjective optimization problems (introduced in Section 2.5) and demonstrated the importance of striking a balance between genetic search and local search. The choice of multiple local learning procedures or memes during a memetic algorithm search in the spirit of Lamarckian learning, otherwise, known as meta-Lamarckian learning, on continuous optimization problems was also considered in Ong et al. [4]. For a detailed taxonomy and comparative study on adaptive choice of memes in memetic algorithms, the reader may refer to [5].

MAs for feature selection have recently been proposed in [44,55] and have shown superior performance over GAs and other search methods. Nonetheless, due to the inefficient nature of sequential LS [44] or random bit climbing LS [55] used in these

existing methods, a large amount of redundant computation is incurred on evaluating the fitness of feature subsets. This makes them less attractive, particularly on problems with large feature size. In this study, we propose novel hybrid filter and wrapper feature selection algorithms using memetic framework and show that filter method based LSs are more efficient and effective.

2.5 Evolutionary Multiobjective Optimization Algorithm

The purpose of feature selection is to reduce the number of features used while maintaining acceptable performance of the induction algorithm. In other words, the objective of feature selection is to minimize the number of selected features and maximize the performance of the induction algorithm at the same time. Obviously the feature selection problem is naturally regarded as a multiobjective optimization problem (MOP).

Without lose of generality, a MOP can be defined as a k -objective minimization¹ problem, i.e.,

$$\min F(s) = (f_1(s), \dots, f_k(s)) \text{ subject to } s \in S \quad (2.2)$$

where $k \geq 2$ and $F(s)$ is the objective vector, $f_i(s)$ is the i -th objective to be minimized, s is the candidate solution vector, and S is the feasible domain of s . Simultaneous optimization of all objectives is not always possible because they can

¹Any maximization problem can be converted to a minimization problem by multiplying with -1.

be conflicting or they measure different aspects of the problem.

The traditional method to deal with MOP is based on function aggregation. The objectives are aggregated into a single and parameterized objective through a linear combination of the objectives, i.e., $\min F(s) = \sum_{i=1}^k w_i f_i(s)$, where $w_i \geq 0$ and $\sum_{i=1}^k w_i = 1$. The weighting coefficients represent the relative importance of the k objective functions. As mentioned in Section 2.2, most of the existing GA based feature selection algorithms use this method by combining and weighting the two objectives of feature selection as a single objective, typically as a linear combination of the generalization error of the induction algorithm and the number of selected features. Such an approach provides the ease of implementation but it has to deal with the issue of identifying an appropriate weight vector, which is usually very difficult and problem-dependent. The obtained optimal solution largely depends on the choice of the underlying weight vector.

Pareto-based multiobjective evolutionary algorithms (MOEAs) [56] represent another approach for handling MOP, which are becoming greater popular than classical weighted sum methods. The core advantage of MOEAs is their ability to find multiple diverse solutions to approximate the Pareto optimal set in a single simulation run, which provides distinct options to solve the problem based on different tradeoffs of the objectives. We begin with some basic concepts of MOEA:

Definition 2.6 *Pareto Dominance:* A vector $u = (u_1, \dots, u_k)$ is said to dominate $v = (v_1, \dots, v_k)$ if and only if u is partially less than v , i.e., $\forall i \in \{1, \dots, k\} : u_i \leq v_i$ and $\exists i \in \{1, \dots, k\} : u_i < v_i$.

Definition 2.7 Pareto Optimal: A solution vector $s^* \in S$ is called Pareto optimal if and only if there is no $s \in S$ for which $u = F(s) = (f_1(s), \dots, f_k(s))$ dominates $v = F(s^*) = (f_1(s^*), \dots, f_k(s^*))$. The set of all Pareto optimal solutions is called the **Pareto Optimal Set** (denoted by $\mathcal{P}^*$).

Definition 2.8 Pareto Front: For a given MOP $\min F(s)$ and the Pareto optimal set $\mathcal{P}^*$, the Pareto Front ($\mathcal{PF}^*$) is defined as: $\mathcal{PF}^* = \{u = F(s^*) = (f_1(s^*), \dots, f_k(s^*)) | s^* \in \mathcal{P}^*\}$

The outline of MOEA is shown in Figure 2.7. We can see that it varies from the canonical EAs in the way the selection operator works, where MOEA has to perform Pareto ranking and maintain the diversity of populations.

MOEA has been successfully applied in solving feature selection problems. For instance, Deb et al. [57] implemented an MOEA for informative genes selection on microarray data classification using Pareto-based method NSGA-II. Banerjee et al. [58] proposed another MOEA for gene selection using rough sets and NSGA-II. MOEAs are also used for feature selection in clustering problems [59, 60].

Similar to single objective EAs, it has been shown that the hybridization of MOEAs and LS can lead to significant improvement in the searching ability of MOEAs [6, 61–66]. Such a hybrid multiobjective algorithm is commonly known as multiobjective memetic algorithm (denoted as MOMA in this study). A number of MOMAs have been successfully applied to various combinatorial problems (e.g., flowshop scheduling problem [6, 62], knapsack problem [63], set-covering problem [65], etc) and obtained a significant improvement on search performance over

Multiobjective Evolutionary Algorithm
BEGIN

1. $t = 0$;
2. **Initialize:** Randomly generate an initial population $P(t)$;
3. Evaluate fitness $F(s)$ of each solution in $P(t)$;
4. Rank the $P(t)$ using Pareto dominance;
5. Distribute $P(t)$ based on diversity measure;
6. **While**(*Termination Criterion Not Fulfilled*)
7. Select a temporary population $P'(t)$ from $P(t)$ based on Pareto rank and diversity measure;
8. Perform recombination and mutation on $P'(t)$;
9. Evaluate fitness $F(s)$ of each solution in $P'(t)$;
10. Rank $\{P(t) \cup P'(t)\}$ using Pareto dominance;
11. Distribute $\{P(t) \cup P'(t)\}$ based on diversity measure;
12. Select solutions from $\{P(t) \cup P'(t)\}$ to create a new population $P(t+1)$ based on Pareto rank and a diversity measure;
13. $t = t+1$;
14. **End While**

END

Figure 2.7: Outline of multiobjective evolutionary algorithm

MOEAs without any LS. In this study, we propose a new MOMA for multiclass feature selection problems, in particular for simultaneous identification of full class and partial class relevant features. More details are described later in Chapter 6.

2.6 Generalization Error Estimation and Overfitting

One key issue in feature selection is how to validate the selected feature subsets. Generally, the fitness of a selected feature subset is evaluated based on the generalization error of the induction algorithm. A simple way to estimate the generalization error of the induction algorithm is to apply the holdout estimator, which splits the whole dataset into separated training and test sets. Such an approach is efficient and provides ease of implementation. Nevertheless, it can be unreliable if the size of the samples in the data is small, since it randomly splits the data only once. In such a case, two more reliable alternative methods, k -fold cross-validation and bootstrap, are used in this dissertation and described in the rest of this section.

2.6.1 k -Fold Cross-Validation

In k -fold cross-validation (CV), the dataset $\mathcal{D}$ is randomly split into k mutually exclusive folds $\mathcal{D}_1, \dots, \mathcal{D}_k$ of equal size. Each fold $\mathcal{D}_i (i \in \{1, \dots, k\})$ is used for testing the induction algorithm and meanwhile its complement, $\mathcal{D}/\mathcal{D}_i$, is used for training. The CV error is then estimated based on the average error over the k test folds. When the sample size is small, the k -fold CV method is always repeated multiple times to reduce the split variance. A special case of k -fold CV is the leave-one-out CV (LOOCV), which has k equal to the sample size of the dataset, such that each time only one sample is left out for testing.

2.6.2 Bootstrap

The bootstrap technique was proposed by Efron and Tibshirani [67]. Given a dataset of size n , a bootstrap sample set is created by randomly sampling n samples with replacement from the dataset. Since the dataset is sampled with replacement, the probability that a sample is not selected across n times sampling is given by $(1 - 1/n)^n \approx e^{-1} \approx 0.368$. The expected number of unique samples in the bootstrap sample set is thus $0.632n$. The induction algorithm is trained on the bootstrap sample set and tested on the set of unselected samples. The .632 bootstrap is repeated k times and the final bootstrap error estimator $\epsilon_{b.632}$ is defined as:

$$\epsilon_{b.632} = \frac{1}{k} \sum_{i=1}^k (0.368\alpha_i + 0.632\beta_i) \quad (2.3)$$

where α_i and β_i denote the training error and test error on the i -th bootstrap, respectively. A variance-reducing technique often employed is the balanced bootstrap resampling [68], where each sample is made to appear exactly k times in the computation. Bootstrap is shown to have lower variance than cross validation on datasets with small sample size [69, 70]. For more details, the reader is referred to [69, 70].

2.6.3 Overfitting

The overfitting problem has drawn increasing attention in the context of feature selection [11, 71–76]. An induction algorithm overfits the dataset if it models the training set overly well which leads the predictions on an unseen test set to be poor.

In feature selection, the overfitting is mainly caused by the high variance of the induction algorithm as well as the selection bias [74,77] of the feature selection method, especially when the sample size of the training set is small or the training set does not fully cover the concept being learned. Moreover, intensive search models and powerful learners would also exacerbate the overfitting problem in feature selection. To alleviate or prevent possible overfitting, a stable induction algorithm should be used. Furthermore, Aliferis et al. [78] also proposed using the theoretical bounds to estimate the generalization error of the induction algorithm, e.g., Radius Margin Bound for estimating the LOOCV error of Support Vector Machines (SVM) [79]. Loughrey et al. [80] suggested to use a early-stopping method to avoid overfitting in wrapper-based feature selection employing stochastic search. In addition, a suitable scheme for generalization error estimation is also critical to avoid reporting overfitting results.

2.6.4 Internal and External Validation

Two schemes for estimating generalization error of the induction algorithm are internal and external validation.

In internal validation, the feature selection method is applied to the entire set of data and the final performance of the induction algorithm is estimated on the same dataset using CV/bootstrap. Although the intention of CV/bootstrap is to provide an accurate estimation of the generalization error, using CV/bootstrap in this manner means that any inference made is done with respect to the induction

algorithm modeling progress. However, because the significant features are usually known at the beginning, both feature selection and induction algorithm learning could possibly result in so called feature subset selection bias [77]. Singhi and Liu [77] defined feature subset selection bias in the context of Bayesian learning and studied in detail the statistical properties of selection bias and various factors that affect the bias. Their study suggests that the current practice of using the same dataset for feature selection and learning is appropriate. Kohavi et al. [71] also show that the problem is not severe on datasets with sufficient samples. However, for datasets with small sample size, the selection bias is not negligible which can lead to increasing chances of overfitting and deteriorate the performance of the induction algorithm. A number of reports [73,74,81,82] appeared in the literature also indicate doubts on the generalization abilities of the reported induction algorithms for feature selection. In [73,74], feature selection methods using internal CV were shown to suffer from overfitting or selection bias on datasets with small sample size.

To address the overfitting problem, the alternative external validation schema was proposed in place of internal validation [73,74]. In such a scheme, feature selection is performed using only the training set which is set aside at each stage of the CV/bootstrap process. The performance of the selected feature subset for the induction algorithm is then evaluated based on the unseen test data. It is worth noting that in wrapper feature selection, another CV/bootstrap could be nested in the external CV/bootstrap. The nested CV/bootstrap, performing only on the training set separated by the external CV/bootstrap, is used to estimate the

significance of the selected candidate feature subset and guide the search for the optimal feature subset.

While, external validation serves to mitigate the effects of overfitting, the tradeoff is a greater computational cost than the internal validation counterpart, especially in wrapper methods. Hence, in practice, one should choose the appropriate scheme for the problem in hand. As a rule of thumb, for datasets with sufficient samples, an internal validation would be preferred for the sake of efficiency. Otherwise, an external validation is more appropriate on datasets with small sample size to avoid the effects of overfitting results.

2.7 Microarray

In this dissertation, our focus of application is biological data, particularly, microarray gene expression data. Here, we give a brief introduction of microarray technology in this section.

DNA microarray [83], a high-throughput and large-scale technology, has attracted tremendous attention from both the scientific and industry communities. By providing snapshots of the expression level of thousands of genes, DNA microarrays promise new insight into fundamental biological phenomena such as development or evolution. DNA microarrays allow simultaneous measurement of activity and interactions of thousands of genes, which qualifies the technology to fulfill various particular scientific tasks including the identification of co-expression genes, discovery of array or gene groups with similar expression pattern, identification of

genes with evidently different expression in term of a set of discerned biological entities (e.g., cancerous tissues), simulation of regulatory gene networks, mapping of expression data to metabolic pathways, etc. On the other hand, they are rapidly producing enormous amounts of experiment data that have never been encountered by scientists before. How to analysis the data on such scale and develop required analytical technologies corresponding to different types of tasks has become a major current challenge faced by scientists.

2.7.1 Microarray Technology

Figure 2.8: Overview of microarray technology

DNA microarray is a glass slide also called gene chip or DNA chip (shown in Figure 2.8). Tens of thousands of DNA molecules are attached on the surface of the chip at fixed spots where each spot contains a DNA special sequence. Two

targets (mRNA samples) are reverse-transcribed into cDNA, which are labeled using different fluorescent dyes (fluors), for instance, a red-fluorescent dye Cy5 and a green-fluorescent dye Cy3. These are then mixed and hybridized with a specific DNA sequence (probes). After this competitive hybridization, the relative abundance of those spotted DNA sequences in two mRNA samples may be assessed by monitoring the differential hybridization (ratio of Cy5/Cy3) of the two samples to the sequences on the array. Afterwards, the slides are imaged using a scanner and fluorescence measurements are made separately for each dye at each spot on the array. There are also other methods to measure the mRNA levels such as Oligonucleotide chips, RT-PCR (Reverse Transcriptase Polymerase Chain Reaction), and SAGE (Serial Analysis of Gene Expression).

2.7.2 Challenges of Analyzing Microarray Data

Recently, a great number of machine learning methods have been used for analyzing microarray data. A significant amount of new discoveries have been made. However, microarray data has remained a great challenge to computer scientists. In particular, the major difficulties lie with the nature of microarray data:

1. Microarray data are highly dimensional with thousands of genes but with only tens or hundreds of samples (arrays). This makes learning from microarray data an arduous task under the effect of curse of dimensionality, where complexity, computational time and required memory resource grow exponentially with the dimension.

2. Microarray data are inherently noisy. The natural biological fluctuations are likely to import measurement variations and bring implications to microarray analysis. In addition, the microarray experiment involves complex scientific procedures during which errors are commonly introduced due to the imperfections of instruments, impurity of materials and negligence by scientists.
3. The biological heterogeneity is another factor that deters the successful analysis of the data. The gene functional classes exhibit great intra heterogeneity due to the difference in derivation organisms and complex regulation systems.
4. Microarray data often contain many irrelevant and redundant genes, which affect the speed and accuracy of most learning algorithms.

Feature selection, also known as gene selection in the context of microarray data analysis, has been widely introduced for addressing the aforementioned problems. In this study, we propose new MA feature selection algorithms for gene selection.

Chapter 3

Wrapper-Filter Feature Selection

Algorithm Using A Memetic

Framework

This chapter presents a novel hybrid wrapper and filter feature selection algorithm for a classification problem using a memetic framework. It incorporates a filter ranking method in the traditional GA to improve classification performance and accelerate the search in identifying the core feature subsets. Particularly, the method adds or deletes a feature from a candidate feature subset based on the univariate feature ranking information. Empirical studies on commonly used datasets from the University of California, Irvine (UCI) repository [23] and microarray datasets show that the proposed method outperforms existing methods in terms of classification accuracy, number of selected features, and computational efficiency. Furthermore, we investigate several major issues of MA to identify a good balance between local search and genetic search so as to maximize search quality and efficiency in the hybrid wrapper and filter MA.

3.1 Introduction

Filter methods are relatively computationally cheap, but they take the risk of selecting subsets of features that may not match the chosen induction algorithm. Wrapper methods generally outperform filter methods in terms of prediction accuracy, but they are generally computationally more intensive. However, wrapper and filter methods can complement each other, in that filter methods can search through the feature space efficiently while the former provides good accuracy. In this chapter, we propose a novel wrapper-filter feature selection algorithm (WFFSA) [22] using a memetic framework [4–7, 44], i.e., a combination of genetic algorithm (GA) [12] and local search (LS). Memetic algorithms (MAs) are population-based metaheuristic search methods inspired by Darwin’s principles of natural evolution and Dawkins’ notion of a meme defined as a unit of cultural evolution that is capable of local refinements. Recent studies on MAs have revealed their success on a wide variety of real-world problems. Particularly, they not only converge to high-quality solutions but also search more efficiently than their conventional counterparts [4–7, 44].

The goal of WFFSA is to improve classification performance and accelerate the search to identify important feature subsets. In particular, the filter method fine-tunes the population of GA solutions by adding or deleting features based on univariate feature ranking information. Hence, our focus here is on filter methods that are able to assess the goodness or ranking of the individual features. We denote such filter methods as filter ranking methods in this study and investigate the proposed WFFSA for several filter ranking methods.

The rest of this chapter is organized as follows. Section 3.2 describes the WFFSA based on a memetic framework. The experimental results and discussions are presented in Section 3.3. Finally, Section 3.4 summarizes this chapter.

3.2 Wrapper-Filter Feature Selection Algorithm

In this section, we introduce the proposed WFFSA for classification problems, which is depicted in Figure 3.1. In the first step, the GA population is initialized randomly, with each chromosome encoding a candidate feature subset. Subsequently, on all or portions of the chromosomes, an LS or meme is applied in the spirit of Lamarckian learning [4–7]. The mechanism to do local improvement is to reach a local optimum or to improve the solution. Genetic operators are then used to generate the next population. This process repeats until the stopping conditions are satisfied. We describe each component in detail as follows.

3.2.1 Encoding Representation and Initialization

In the feature selection problem, a representation for a candidate feature subset must be chosen and encoded as a chromosome. In most studies, a chromosome is a binary string of length equal to the total number of features, so that each bit encodes a single feature (as shown in Figure 3.2). A bit of ‘1’ (‘0’) implies the corresponding feature is selected (excluded). The length of the chromosome is denoted here as n . The maximum allowable number of 1’s in each chromosome is denoted as m . When prior knowledge about the optimal number of features is available, we may limit m

Procedure of WFFSA
BEGIN

1. **Initialize:** Randomly generate an initial population of feature subsets encoded with binary strings;
2. **While** (*Termination Criterion Not Fulfilled*)
 3. Evaluate fitness of all feature subsets in the population base on $J(S_c)$;
 4. **For** each solution chosen to undergo the local improvement process
 5. Perform filter ranking based local search and replace it with a locally improved solution in the spirit of Lamarckian learning;
 6. **End For**
 7. Perform evolutionary operators based on selection, crossover, and mutation;
8. **End While**

END

Figure 3.1: The Procedure of WFFSA

to no more than the predefined value; otherwise, m is equal to n . At the start of the search, a population size of p is randomly initialized.

Figure 3.2: Representation of a chromosome as a binary string.

3.2.2 Objective Function

The objectives of feature selection are to maximize the classification accuracy and meanwhile minimize the number of selected features. As discussed in Section 2.2 and Section 2.5, one way to handle this problem is to use the aggregating function method (as shown in Equation 2.1), where the objective function is defined as a linear combination of the generalization error and the number of selected features. The weights can be empirically tuned based on the bounds of the number of selected features and the generalization error. However, the weights would significantly vary for different datasets and the selected feature subset would largely depends on the underlying weight vector.

To avoid the aforementioned problem, the classification accuracy is considered to be more important than the selected feature size in this study. The objective function is defined by the generalization error of the induction algorithm, i.e.,

$$Fitness(c) = J(S_c) \quad (3.1)$$

where S_c denotes the corresponding selected feature subset encoded in chromosome c , and the feature selection criterion function $J(S_c)$ is specified as the classification accuracy for S_c . When two chromosomes are found having similar fitness (i.e. for a misclassification error of less than one data instance, the difference between their fitness is designed here to be less than a small value of $\varepsilon = 1/D$, where D is the number of instances) the one with a smaller number of selected features is given a higher chance of surviving to the next generation.

3.2.3 Local Search Improvement Procedure

Many studies on the use of domain knowledge and heuristics have resulted in highly effective search [4–6]. Taking this cue, we consider here the use of filter ranking methods as memes or LS heuristics in our WFFSA. Later, we show in Section III that using filter ranking methods as memes, MA is capable of converging to improved classification accuracy and lower number of selected features when compared to existing methods recently proposed in the literature.

Given a candidate chromosome c , we define X and Y as the sets of selected and excluded features encoded in c , respectively. Each feature in both X and Y is ranked using the univariate filter ranking method and with the most important feature ranked the highest. In this study, we consider three different filter ranking methods, namely: 1) ReliefF [24]; 2) Gain Ratio [25]; and 3) Chi-square [26] (For more details of these methods, please refer to Appendix A). These methods rank features based on different criteria that include Euclidean distance, information entropy, and chi-square statistics, respectively. We further define two basic LS operators of the WFFSA LS improvement procedure.

1. Add: select a feature from Y using the linear ranking selection [84] and move it to X .
2. Del: select a feature from X using the linear ranking selection [84] and move it to Y .

The Add and Del operations are illustrated in Figure 3.3. Here, $F5$ and $F4$ are the highest and lowest ranked features in Y , respectively, whereas $F3$ and $F6$ are

the highest and lowest ranked features in X , respectively. Using the Add and Del operations, $F5$ is the most likely feature to be moved to X , whereas $F6$ is the most likely feature to be moved to Y . The two most likely resultant chromosomes after the Add and Del operations are also depicted in Figure 3.3.

Figure 3.3: Add and Del operations (X and Y denote the selected and excluded feature subsets, respectively).

We define the maximum number of Del and Add operations each as the LS length, l . Therefore, there are a total number of l^2 possible combinations of Add and Del operations applied on a chromosome for each LS. In each generation, w , denoted as LS interval, elite chromosomes in the population are selected to undergo LS improvement procedure in each generation. The LS improvement procedure may be applied on a chromosome until a local optimum or an improvement is reached. The locally improved chromosome is evaluated and replaces the original chromosome if it is of higher quality. Here, we further investigate three different LS strategies

that are characterized by different LS intensity.

Improvement First Strategy

In this strategy, a random choice from the l^2 combinations of Del and Add operations is used to search on the candidate chromosome. The LS stops once an improvement is obtained either in terms of classification accuracy or a reduction in the number of selected features without deterioration in accuracy greater than ε . This procedure is outlined in Figure 3.4.

Steepest Descent Strategy

In contrast to the improvement first strategy, the steepest descent strategy carries out all possible l^2 combinations of Del and Add operations, and the best improved solution is used to replace the original chromosome in the population. The steepest descent strategy is outlined in Figure 3.5.

Sequential Strategy

We also consider the sequential strategy described in [44]. The hybrid GA (HGA) was reported in [44] to generate better search performances than GA, sequential forward search (SFS), sequential forward floating search (SFFS), PTA(k,d) (plus k and take away d) and multistart algorithms. There, instead of using a filter ranking method, the Add operation searches for the most significant feature y in Y in a sequential manner, i.e., $y = \arg \max_{a \in Y} J(X \cup \{a\})$, and moves it to X . In the same way, the Del operation searches for the least significant feature x from X in a

Procedure of Improvement First Strategy
BEGIN

1. Initialize the LS length l and interval w ;
2. **For** each chromosome c among the w elitists
3. **For** ($j = 1$ to l^2)
4. Generate an unique random pair (k, d) where $0 \leq k, d < l$;
5. Repeat k times of Add operation;
6. Repeat d times of Del operation;
7. Calculate fitness of improved chromosome c' using $F(c') = J(c')$;
8. **If** $((F(c') > F(c)))$ **or** $(|F(c') - F(c)| < \varepsilon)$ **and** $|c'| < |c|$)

//|c| denotes the cardinality of c
9. Replace the genotype c with the improved c' ;
10. Break and consider the next elite chromosome;
11. **End If**
12. **End For**
13. **End For**

END

 Figure 3.4: Procedure of improvement first strategy.

sequential manner, i.e., $x = \arg \max_{a \in X} J(X - a)$, and moves it to Y .

3.2.4 Evolutionary Operators

In the evolution process, standard GA operators such as linear ranking selection, uniform crossover, and mutation operators based on an elitist strategy may be applied. However, if prior knowledge on the optimum number of features is available, the number of 1's in each chromosome may be constrained to a maximum of m in

Procedure of Steepest Descent Strategy
BEGIN

1. Initialize the LS length l and interval w ;
2. **For** each chromosome c among the w elitists
3. $c_{best} = c$;
4. **For** each of the l^2 combinations (k, d) where $0 \leq k, d < l$
5. Repeat k times of Add operation;
6. Repeat d times of Del operation;
7. Calculate fitness of improved chromosome c' using $F(c') = J(c')$;
8. **If** $((F(c') > F(c)))$ **or** $(|F(c') - F(c)| < \varepsilon)$ **and** $|c'| < |c|$
9. $c_{best} = c'$; *//Update the best improved chromosome*
10. **End If**
11. **End For**
12. Replace the genotype c with c_{best} ;
13. **End For**

END

Figure 3.5: Procedure of steepest descent strategy.

the evolution process. Since the standard uniform crossover and mutation operators may violate this constraint, restrictive crossover and mutation are proposed here. In restrictive crossover, the crossover operations are applied only on alleles with bit 1 in either of the two parent chromosomes. We outline the restrictive crossover in Figures 3.6 and 3.7. Figure 3.7 depicts an example of alleles to undergo the restrictive crossover operations on two parent chromosomes. Based on the same principle, the restrictive mutation operator is outlined in Figure 3.8.

Procedure of Restrictive Crossover.**BEGIN**

1. Randomly select two parents p_1 and p_2 ;
2. Randomly generate a number r within $[0, 1]$;
3. **If** ($r < p_c$) *// p_c denotes the crossover probability*
// ensure the number of bit '1' in offspring dose not exceed m ;
 4. $k = \min(|p_1|, |p_2|)$; *// $|p_1|, |p_2| < m$*
 5. **For** ($i = 1$ to k)
 6. Locate the allele L_1 of the i -th bit '1' in p_1 ;
 7. Locate the allele L_2 of the i -th bit '1' in p_2 ;
 8. Crossover p_1 and p_2 in position L_1 and L_2 with probability 0.5;
 9. **End For**
10. **End If**

END

Figure 3.6: Procedure of restrictive crossover.

Figure 3.7: An example of alleles to undergo restrictive crossover operation.

3.2.5 Computational Complexity

In this section, we analyze the computational complexity of the proposed WFFSA. The ranking of features based on the filter methods have linear time complexity in terms of feature dimensionality. They are conducted offline, and the obtained rank list may be reused for each LS in WFFSA. Consequently, the computational cost for

Procedure of Restrictive Mutation.**BEGIN**

1. **For**($i = 1$ to $|c|$)
2. Locate the position L_1 of the i -th bit '1' in c ;
3. Randomly select a bit '0' with position L_0 ;
4. Swap position L_0 and L_1 with probability p_m ;
 // p_m denotes the mutation probability;
5. **End For**
6. **For**($i = 1$ to $m - |c|$)
7. Randomly flip a bit '0' with probability p_m ;
8. **End For**

END

Figure 3.8: Procedure of restrictive mutation.

feature ranking is a one-time offline cost and is considered to be negligible compared to that of fitness evaluation in Equation 3.1. Hence, we define the computational cost of a single fitness evaluation as the basic unit of computational complexity in our analysis.

The computational complexity for GA can be derived as $O(pg)$, where p is the size of the population and g is the number of search generations. The expected computational complexity of WFFSA with the improvement first strategy is $O(pg + l^2wg/2)$. We note that, in the early stage of LS, many neighbors have better fitness values than the current solution, therefore, LS with improvement first strategy tends to identify the improved solution and stop the search very quickly. In the final stage of LS, the current solution generally has fewer improved neighbors. Hence, LS has

to search across all possible l^2 neighbors exhaustively. Here, we approximate the average trails to obtain the first improvement is $l^2/2$. In a single search generation, $l^2w/2$ fitness function calls are incurred in the LS. The ratio of LS to genetic search is thus $l^2w/2p$.

In the steepest descent strategy, LS evaluates all possible l^2 combinations of Add and Del operations to attain the best possible locally improved solution. The computational complexity is thus $O(pg + l^2wg)$. In a single search generation, l^2w fitness function calls are incurred in the LS, and the ratio of LS to genetic search is l^2w/p .

For the sequential strategy proposed in HGA [44], a total of $(2|Y| - l)l/2$ and $(2|X| + l)l/2$ calls to the fitness function are incurred on the Add and Del operations, respectively. Since $|X| + |Y| = n$, the total classification calls in the LS is $n lw$. The computational complexity of the sequential strategy is thus $O(pg + n lwg)$. The ratio of LS to genetic search is then $n lw/p$.

Since we are working with classification problems where $n \gg l$, it is easy to determine that $n lw \gg l^2w > l^2w/2$. Consequently, the sequential LS strategy in HGA [44] would require significantly more computations, hence incurring more time than both the improvement first strategy and steepest descent strategy.

3.3 Experimental Results and Discussions

In this section, we present an experimental study of WFFSA on commonly used benchmark and biological datasets. In particular, we focus on four UCI datasets

having more than 15 features and four microarray datasets, ALL-AML1000 [85], Colon1000 [85], NCI1000 [16], and SRBCT [86], (for more details about these microarray datasets, the reader is referred to Appendix D) having thousands of features in our study.

In the WFFSA, we employed a population size of 30 and a stopping criterion of 6000 fitness function calls for UCI datasets. On microarray datasets, as the feature size is significantly larger, the population size is increased to 50 or 100, and the stopping criterion is increased to 10 000 or 20 000. Furthermore, the maximum number of selected features is unconstrained for the UCI datasets, i.e., $m = n$. On the other hand, it has been shown in the literature [16, 85–88] that microarray datasets could be learned with high accuracies with only hundreds or tens of features. In effect, we make use of this prior knowledge to constrain the maximum number of selected features m as depicted in Table 3.1. In effect, the restrictive crossover and mutation operations are applied on the microarray datasets. In our experimental setup, we employ crossover and mutation probabilities of $p_c = 0.6$ and $p_m = 0.1$, respectively. The threshold ε to determine fitness similarity between two chromosomes is configured as 0.001 for UCI datasets and 0.02 for microarray datasets. The fitness of a chromosome or selected feature subset is evaluated using the k-nearest neighbor (kNN) classifier with $k = 1$ and the LOOCV. Here, we use the classification accuracy estimated from LOOCV and the number of selected features as performance measures. It is worth noting that the configurations of the parameters used here have been investigated empirically for the datasets considered and are summarized

in Table 3.1.

Table 3.1: Datasets and Parameters Used for Experiments

Dataset	# of Features	# of Instances	# of Classes	Population Size	p_c, p_m	Max # of Selected Features	Stopping Criterion*
Vehicle	18	846	4	30	0.6,0.1	18	6000
WDBC	30	569	2	30	0.6,0.1	30	6000
Ionosphere	34	351	2	30	0.6,0.1	34	6000
Sonar	60	208	2	30	0.6,0.1	60	6000
ALL-AML1000	1000	72	2	50	0.6,0.1	Constrained to 50	10000
Colon1000	1000	62	2	50	0.6,0.1	Constrained to 50	10000
NCI1000	1000	60	9	50	0.6,0.1	Constrained to 50	10000
SRBCT	2308	83	4	100	0.6,0.1	Constrained to 50	20000

*Here the stopping criterion is the maximum number of fitness function calls.

3.3.1 Comparison of Filter, GA, and WFFSA

The search performances of the filter ranking methods GA and WFFSA using several filter ranking methods and LS strategies on the eight datasets considered are summarized in Table 3.2. The comma-delimited pairwise numeric values in the table represent the classification accuracy and the corresponding number of selected features. Due to the stochastic nature of GA and WFFSA, the average results for ten independent runs are reported. The parentheses highlight the best result found across the ten runs. For each dataset, the bold-faced and bold-italic-faced representations in Table 3.2 emphasize the best average performance and the best solution found among all methods, respectively. Six WFFSAs obtained based on the three filter ranking methods (i.e., ReliefF—WFFSA-R, Gain Ratio—WFFSA-G, or Chi-square—WFFSA-C) and two LS strategies (i.e., improvement first or steepest descent) have been investigated. In the WFFSAs, the LS length l and interval w in the improvement first strategy and steepest descent strategy are configured as $l = 4$,

$w = 1$ and $l = 4$, $w = 5$, respectively.

We discuss the performances for the various feature selection algorithms considered. On filter ranking methods, the t features with highest rank are chosen to induce the 1NN LOOCV classification. We increase t from 1 to m , and the optimal value is determined when the best classification accuracy is obtained. The best classification accuracy and the corresponding t value for each filter ranking method are reported in Table 3.2 (i.e., columns 3-5). It can be observed that WFFSAs outperform all three filter ranking methods and GA in terms of classification accuracy. The improvement in performance is more significant for the Sonar, Colon1000, and NCI1000 datasets. Moreover, WFFSAs reduce the number of selected features significantly. On the ALL-AML1000 and SRBCT datasets, WFFSAs use less than one-third of the features required by GA and the filter ranking methods to arrive at the improved classification accuracy. The best solutions found on all eight datasets shown in column 2 were attained by WFFSAs. We also study the performance of WFFSAs for the three different filter ranking methods since it was shown in [4] that inappropriate use of meme may result in MA performing poorer than standard GA. The results in Table 3.2 indicate that WFFSA-R performs better than the other counterparts on seven out of the eight datasets in terms of average performance. For ReliefF filter ranking method, the feature goodness is evaluated by its ability to distinguish the near hit (nearest neighbors from the same class) and near miss (nearest neighbors from different classes). Therefore, it makes good sense that features with high score in ReliefF are more likely to help 1NN identify the correct nearest

neighbor, hence generating good classification accuracy. On the other hand, Gain Ratio and Chi-square do not appear to possess such mechanisms to compliment 1NN, hence being less effective here.

Furthermore, we compare the results obtained by WFFSAs to the recently proposed HGA(3) [44] for the UCI datasets in Table 3.3. The best average performance and the best solution for each dataset are highlighted using bold typeface representation. The results indicate that WFFSAs generate the best solutions on all four datasets and give competitive results to existing methods in terms of average performance. Even so, it is worth noting that HGA(3) consumed more than 200 000 fitness function calls to arrive at the competitive performance on these UCI datasets. In contrast, WFFSA incurs less than 6000 fitness function calls to arrive at superior or competitive performances.

To illustrate the generality and efficacy of the WFFSA framework, we consider also the use of different induction algorithms, particularly standard 1NN and support vector machine (SVM) with radius margin bound [79] (details given in Appendix B), and compare their performances with other recent studies in the literature using internal and external validation schemes on the real-world microarray datasets. In external cross validation, each dataset is randomly split into k stratified folds with $k - 1$ folds for training and one fold for testing. The performance of the final selected features obtained is measured on the unseen testing data. The procedure is repeated k times. The best average performance and/or the best solution are reported in Table 3.4. Table 3.4 indicates that WFFSAs display superior performances on most

Table 3.2: Comparison Results of Filter, GA, and WFFSA Methods

Dataset	Best Found	ReliefF	Gain Ratio	Chi Square	GA	WFFSA-R*		WFFSA-G*		WFFSA-C*	
						First**	Steepest**	First	Steepest	First	Steepest
Vehicle (846×18)	75.06,12	72.81,12	70.33,16	72.22,15	74.88,10.2 (75.06,12)	75.00,11.3 (75.06,12)	75.06,12 (75.06,12)	74.98,11.5 (75.06,12)	75.04,11.7 (75.06,12)	75.06,12 (75.06,12)	75.04,11.7 (75.06,12)
WDBC (569×30)	98.24,12	96.49,4	96.31,29	96.31,29	97.82,14.7 (98.23,19)	97.96,13 (98.24,14)	98.14,13.9 (98.24,14)	97.98,11.8 (98.24,14)	97.93,13.2 (98.07,11)	97.88,12 (98.24,15)	98.00,12.7 (98.24,12)
Ionosphere (351×34)	96.01,8	91.74,7	90.88,13	92.31,7	94.13,12.5 (94.87,13)	95.00,7.5 (95.73,8)	94.96,9.7 (95.44,8)	95.13,8.9 (96.01,8)	95.19,8.9 (95.44,7)	95.01,8.9 (95.44,7)	94.73,10.4 (95.16,9)
Sonar (208×60)	97.6,22	88.46,15	88.94,31	89.90,30	93.89,27.7 (95.19,25)	96.30,24 (97.12,19)	95.63,24.6 (96.63,24)	94.95,25.2 (97.6,22)	95.22,24 (96.63,24)	95.82,24.6 (96.63,24)	95.24,26.4 (96.15,26)
ALL-AML1000 (72×1000)	100,2	98.61,16	98.61,17	98.61,18	100,26.5 (100,20)	100,5.1 (100,2)	100,8.3 (100,4)	100,5.5 (100,3)	100,6.3 (100,4)	100,6.1 (100,2)	100,7.9 (100,3)
Colon1000 (62×1000)	100,14	83.87,17	90.32,24	87.10,17	94.52,37 (95.16,31)	97.9,10.9 (100,14)	97.26,16 (100,20)	96.29,11.9 (98.39,11)	97.9,14.9 (100,17)	95.97,10 (98.39,15)	96.94,17.8 (100,16)
NC11000 (60×1000)	85.25,14	65.57,16	59.02,31	65.57,28	77.54,38.6 (80.33,40)	82.30,22.7 (85.25,19)	79.67,23.8 (81.97,24)	82.13,21.5 (85.25,18)	81.31,27.5 (85.25,33)	81.15,21.8 (85.25,14)	81.48,26.5 (85.25,22)
SRBC-T (83×2308)	100,7	100,38	100,20	100,43	99.64,43.3 (100,39)	100,15.1 (100,7)	100,18.7 (100,11)	99.76,17.1 (100,8)	100,21.8 (100,16)	99.76,14.9 (100,9)	99.88,17.9 (100,8)

The value delimited by comma in each grid shows the classification accuracy and the corresponding number of selected features respectively. The values in the parentheses are the best results obtained. Bold typefaces emphasize the best average performance in each row. Bold italic typefaces emphasize the best solution found among all the methods.

* Here WFFSA-R, WFFSA-G, and WFFSA-C denote WFFSA with local search heuristic of ReliefF, Gain Ratio, and Chi-Square, respectively.

** First represents the improvement first strategy with length $l = 4$ and interval $w = 1$. Steepest represents steepest descent strategy with $l = 4$ and $w = 5$.

Table 3.3: Results by WFFSAs and Other Methods on UCI Datasets

Dataset	WFFSA		Results Obtained From the Literature	
Vehicle	75.06,12 (75.06,12)	WFFSA+1NN LOOCV	73.52,7 (73.52,7)	HGA(3)+1NN LOOCV [44]
WDBC	98.14,13.9 (98.24,12)	WFFSA+1NN LOOCV	93.85,24 (93.85,24)	HGA(3)+1NN LOOCV [44]
Ionosphere	95.19,8.9 (96.01,8)	WFFSA+1NN LOOCV	95.56,7 (95.73,7)	HGA(3)+1NN LOOCV [44]
Sonar	96.30,24 (97.6,22)	WFFSA+1NN LOOCV	96.34,24 (97.12,24)	HGA(3)+1NN LOOCV [44]

of the microarray datasets in comparison to the existing counterparts.

Table 3.4: Results by WFFSAs and Other Methods on Microarray Datasets

Dataset	WFFSA		Results Obtained From the Literature	
ALL-AML1000(EV*)	97.34,30.5 (98.57,30.8)	WFFSA+SVM 10-fold ECV*	97.34,31 (97.68,50)	SVM+LS bound .632+ bootstrap [85]
ALL-AML1000(IV*)	100,5.1 (100,2)	WFFSA+1NN LOOCV	(100,8)	SVM-RFE Holdout [88]
Colon1000(EV)	86.01,31.0 (87.38,31.2)	WFFSA+SVM 10-fold CV	84.77,31 (84.95,46)	SVM+LS bound .632+ bootstrap [85]
Colon1000(IV)	97.9,10.9 (100,14)	WFFSA+1NN LOOCV	(100,16)	SVM-RFE Holdout [88]
SRBCT(EV)	98.53,90.5 (100,89.6)	WFFSA+SVM 10-fold CV	(95.00,150)	SVM+Max Minority 4-fold CV [87]
SRBCT(IV)	100,15.1 (100,7)	WFFSA+1NN LOOCV	(100,96)	ANN+PCA 3-fold CV [86]
NCI1000(EV)	66.43,485.3 (72.19,485)	WFFSA+SVM 4-fold CV	(66.66,150)	SVM+Sum Minority 4-fold CV [87]
NCI1000(IV)	82.30,22.7 (85.25,14)	WFFSA+1NN LOOCV	(85.25,13)	GA+MLHD LOOCV [16]

* EV: External Validation; IV: Internal Validation

3.3.2 Study on LS Strategies

Over the recent years, much work has shown that an appropriate balance between LS and genetic search is necessary for efficient memetic search [4–7]. In this section, we examine the balance between LS and genetic search using different LS strategies, which are characterized by different intensities of genetic search and LS. Ten independent runs of WFFSA with different combinations of strategy, LS length l , and interval w are conducted on two representative UCI and microarray data sets, i.e.,

the Sonar and Colon1000 datasets. The results obtained are summarized in Tables 3.5 and 3.6. The LS heuristic or filter ranking method used is ReliefF since it is shown to give better search performance than the other counterparts.

Table 3.5: Results of Different LS Strategies, Length and Interval on Sonar Dataset

	Improvement First Strategy			Steepest Descent Strategy		
	$w = 1$	$w = 5$	$w = P$	$w = 1$	$w = 5$	$w = P$
$l = 2$	95.77,26.3	94.95,26.7	95.10,26.1	94.86,28.3	95.14,25.3	94.52,28.3
$l = 4$	96.30,24	95.67,24.7	95.38,22.9	94.8,23.2	95.63,24.6	94.13,26
$l = 8$	95.53,21.4	95.24,24.4	95.43,25.5	94.91,20.5	95.10,25	94.04,28.1

Table 3.6: Results of Different LS Strategies, Length and Interval on Colon1000 Dataset

	Improvement First Strategy			Steepest Descent Strategy		
	$w = 1$	$w = 5$	$w = P$	$w = 1$	$w = 5$	$w = P$
$l = 2$	97.26,17.9	97.10,16.9	95.97,22.2	96.13,15	96.29,14.7	95.65,20.8
$l = 4$	97.9,11.5	97.42,12.9	97.10,14.6	96.45,12.6	97.26,16	95.65,19.9
$l = 8$	96.77,10.5	97.26,10.4	97.10,13.1	96.45,10.9	96.94,14.1	95.32,15.8

The results in Tables 3.5 and 3.6 indicate that WFFSA for $l = 4$ and $w = 1$ obtains the best average accuracy on the improvement first strategy. On the Sonar dataset, WFFSA for $l = 4$ and $w = 1$ and using the improvement first strategy is found to perform significantly better than the other configurations of WFFSAs statistically, using the two-tailed paired t-test at significance level of 0.05. On the Colon1000 dataset, the superiority of WFFSA for $l = 4$, $w = 1$ and an improvement first strategy is, however, not statistically conclusive using the same t-test. Overall, we note that WFFSA with improvement first strategy generally performs better than using the steepest descent strategy. WFFSA obtains better results when LS is

applied only to the single or five elite chromosomes instead of the entire population. In line with the observations in [4] and [7], conducting LS on all chromosomes can result in unnecessary computation. Since the computational budget is often limited (i.e., based on the maximum number of fitness function calls allowed), the genetic search will be reduced proportionally due to possible redundant computation spent on LS. Without sufficient genetic search, the MA is more likely to be trapped in the local optimum and hence may lead to poor search quality under a limited computational budget.

We also compare WFFSA-R with an MA based on sequential strategy (MA-S). In MA-S, we configure $l = 4$ and $w = 5$. The comparison results are reported in Table 3.7. On all the datasets, both WFFSA-Rs with improvement first strategy or steepest descent strategy (they are labeled as WFFSA-R First and WFFSA-R Steepest, respectively, in Table 3.6) display significantly better performance than MA-S statistically using two-tailed paired t-test at significance level of 0.05. The superior performance of WFFSA over MA-S is more obvious on the microarray datasets where $n \gg l$. So far, the results obtained further strengthen the importance of the balance between LS and genetic search for efficient MA search. A search dominated by either genetic search (e.g., GA) or LS (MA-S) would generally not perform effectively. The WFFSA for $l = 4$ and $w = 1$ based on the improvement first strategy thus gives the most appropriate tradeoff of LS and genetic search than GA, MA-S, and all other configurations considered in this study. As shown in Figure 3.9-3.16, the average search trends of WFFSA-R First, WFFSA-R Steepest, MA-S

($l = 4, w = 1$), and MA-S ($l = 4, w = 5$) on all datasets suggest that WFFSA-Rs search more efficiently than GA and MA-S under limited computational budget.

Table 3.7: Results of Different LS Strategies on All the Eight Datasets

Dataset	WFFSA-R First $w = 1, l = 4$	WFFSA-R Steepest $w = 5, l = 4$	MA-S* $w = 1, l = 4$	MA-S* $w = 5, l = 4$
Vehicle (846×18)	75.00,11.25 (75.06,12)	75.06,12 (75.06,12)	74.84,10.5 (75.06,12)	74.60,10.6 (75.06,12)
WDBC (569×30)	97.96,13 (98.24,14)	98.14,13.9 (98.24,14)	97.80,14.7 (98.07,14)	97.47,13.7 (97.72,12)
Ionosphere (351×34)	95.00,7.5 (95.73,8)	94.96,9.7 (95.44,8)	94.27,11 (95.44,10)	93.48,12.8 (94.59,12)
Sonar (208×60)	96.30,24 (97.12,19)	95.63,24.6 (96.63,24)	93.65,26.5 (95.67,25)	91.73,28.4 (93.75,32)
ALL-AML1000 (72×1000)	100,5.1 (100,2)	100,8.3 (100,4)	100,27.2 (100,16)	100,44 (100,44)
Colon1000 (62×1000)	97.9,11.5 (100,14)	97.26,16 (100,20)	96.77,48 (98.39,13)	95.17,13 (95.17,13)
NC1000 (60×1000)	82.30,22.7 (85.25,19)	79.67,23.8 (81.97,24)	76.89,36.9 (78.69,38)	75.41,36 (75.41,36)
SRBCT (83×2308)	100,15.1 (100,7)	100,18.7 (100,11)	100,78 (100,78)	98.8,35 (98.8,35)

* MA-S denotes memetic algorithm based on sequential local search strategy.

3.4 Summary

In this chapter, we have proposed a novel hybrid filter and wrapper feature selection algorithm, WFFSA, based on a memetic framework. We use a filter ranking method as an LS heuristic in the MA. The experimental results presented show that, WFFSA searches more efficiently with the filter method fine-tuning the population of GA solutions and is capable of producing good classification accuracy with a small number of features simultaneously. Most importantly, it outperforms GA, MA with

Figure 3.9: Search trace on Vehicle dataset.

Figure 3.10: Search trace on WDBC dataset.

Figure 3.11: Search trace on Ionosphere dataset.

Figure 3.12: Search trace on Sonar dataset.

Figure 3.13: Search trace on ALL-AML1000 dataset.

Figure 3.14: Search trace on Colon1000 dataset.

Figure 3.15: Search trace on NCI1000 dataset.

Figure 3.16: Search trace on SRBCT dataset.

sequential LS, as well as many existing algorithms in the literature. Furthermore, our study on various LS strategies, LS length, and interval allows us to identify a suitable balance of genetic search and LS in the memetic search. This allows us to maximize the effectiveness and efficiency of the proposed hybrid filter and wrapper feature selection algorithm for classification problem using a memetic framework.

Chapter 4

Markov Blanket-Embedded

Genetic Algorithm for Gene

Selection

In Chapter 3, we proposed a hybrid wrapper and filter feature selection algorithm, WFFSA, using MA. It has shown promising performance on high dimensional data. However, because the filter methods (i.e., ReliefF, Gain Ratio, and Chi-square) used in WFFSA are forms of univariate ranking approaches, they generally fail in identifying possible interactions between features and hence may not be capable of identifying redundant features efficiently. In this chapter, we propose a novel Markov blanket-embedded genetic algorithm (MBEGA) [29] targeting to handle gene selection problems with a large number of irrelevant and redundant features. In particular, the embedded Markov blanket based memetic operators add or delete features (or genes) from a GA solution so as to quickly improve the solution and fine-tune the search. Empirical results on synthetic and microarray benchmark datasets suggest that MBEGA is effective and efficient in eliminating irrelevant and

redundant features based on both Markov blanket and predictive power in a GA wrapper model. A detailed comparative study with other methods from each of filter, wrapper, and standard GA shows that MBEGA gives a best compromise among all four evaluation criteria, i.e., classification accuracy, number of selected genes, computational cost, and robustness. The comprehensive comparison between WFFSA and MBEGA will be investigated in next chapter.

4.1 Introduction

Recently, genome-wide microarray gene expression data from cancerous tissues has provided clues for cancer classification and accurate diagnostic tests. Machine learning techniques have been successfully applied to cancer classification using microarray data [13–19, 45, 74, 76, 85, 87–92]. A significant amount of new discoveries have been made and new biomarkers for various cancers have been detected from the microarray data analysis. However, cancer classification has remained a great challenge to computer scientists due to the noisy and high-dimensional nature of the microarray gene expression data. The inherent difficulties on analyzing microarray data, as detailed in Section 2.7.2, make cancer classification an arduous task. Feature selection, also known as gene selection in the context of microarray data analysis, addresses the aforementioned problems by removing the irrelevant and redundant features.

Traditional gene selection methods often uses filter scheme. They rank genes according to their individual relevance or discriminative power with respect to the

target classes and select the top ranked genes [89, 91]. Most of these filter methods are forms of univariate ranking approaches, hence they are not efficient in identifying possible interactions between features and may not be capable of identifying redundant features. Furthermore, since they do not involve the induction algorithm, they are subjected to the risk of selecting subsets of features which may not match the chosen induction algorithm. Wrapper methods, on the contrary, use the induction algorithm itself to evaluate the candidate gene subsets. They generally select gene subsets more suitable for the induction algorithm than the filter methods, but they are generally computationally more intensive

On the other hand, a key issue for gene selection algorithm is how to search the space of feature subsets which is exponential in the extremely huge number of genes. GA [12] has well-known ability to produce high quality solutions within tractable time even on complex problems. It has been naturally used for gene selection and has shown promising performance [13–21] (for a survey of GA-based methods on microarray analysis, the reader is referred to [45]). Unfortunately, due to the inherent nature of GA, it often takes a long time to locate the local optimum in a region of convergence and sometimes may not find the optimum with sufficient precision.

To address the aforementioned problems, we present a novel MA using possible synergy between Markov blanket filter and GA wrapper methods in this chapter, particularly, a Markov blanket embedded GA (MBEGA) feature selection algorithm for cancer classification problems. MBEGA uses Markov blanket to fine-tune the

search by adding the relevant features or removing the redundant and/or irrelevant features in the GA solutions. This MA takes advantage of both Markov blanket and GA wrapper feature selection with the aim to improve classification performance and accelerate the search to retain relevant and remove redundant features. We take representative methods from each of filter, wrapper, and standard GA and show that the performance of MBEGA is comparable to or superior to the existing counterparts in terms of classification accuracy, number of selected features, computational cost, and/or robustness.

The rest of this chapter is organized as follows. Section 4.2 describes our proposed algorithm MBEGA. Section 4.3 presents experimental results and discussions on four synthetic datasets and eleven microarray datasets. Finally, Section 4.4 concludes this chapter.

4.2 System and Methodology

In this section, we present an overview of Markov Blanket, Approximate Markov Blanket and our proposed Markov Blanket Embedded GA (MBEGA) feature selection method.

4.2.1 Markov Blanket

In 1996, Koller and Sahami [27] proposed a cross-entropy based feature selection method and Markov Blanket [31] for identifying redundant and irrelevant features. If F is the full set of features and C the class, the Markov blanket of a feature F_i is

defined as follows:

Definition: (Markov Blanket) Let M be a subset of features which does not contain F_i , i.e., $M \subseteq F$ and $F_i \notin M$. M is a Markov blanket of F_i if F_i is conditionally independent of $(F \cup C) - M - \{F_i\}$ given M , i.e., $P(F - M - \{F_i\}, C | F_i, M) = P(F - M - \{F_i\}, C | M)$.

Two attributes A and B are conditionally independent given K , if $P(A|K, B) = P(A|K)$, that is B gives no information about A beyond what is already in K . If a feature F_i has a Markov blanket M within the currently selected feature subset, it suggests that F_i gives no more information beyond M about C and other selected features, therefore, F_i could be removed safely. However, since the computational complexity to determine the conditional independence of features is typically very high, Yu and Liu [28, 90] considered using only one feature to approximate the Markov blanket of F_i .

Definition: (Approximate Markov Blanket) For two features F_i and $F_j (i \neq j)$, F_j is said to be an approximate Markov blanket of F_i if $SU_{j,C} \geq SU_{i,C}$ and $SU_{i,j} \geq SU_{i,C}$ where the symmetrical uncertainty SU [93] measures the correlation between features (including the class, C). It is defined as:

$$SU(F_i, F_j) = 2 \left[\frac{IG(F_i|F_j)}{H(F_i) + H(F_j)} \right] \quad (4.1)$$

where $IG(F_i|F_j)$ is the information gain [25] between the features F_i and F_j , $H(F_i)$ and $H(F_j)$ denote the entropies of F_i and F_j respectively. $SU_{i,C}$ denotes the correlation between the feature F_i and the class C , and is named *C-correlation*. A

feature is thus considered to be relevant if its C -correlation is higher than some given threshold γ which is user-specific, i.e., $S_{i,C} > \gamma$. A relevant feature without any approximate Markov blanket is called a predominant feature.

4.2.2 Markov Blanket Embedded GA

In this section, we give details on the proposed gene selection method, Markov Blanket Embedded GA (MBEGA). The pseudo code of the MBEGA is outlined in Figure 4.1.

Markov Blanket Embedded Genetic Algorithm (MBEGA)

BEGIN

1. **Initialize:** Randomly generate an initial population of feature subsets encoded with binary strings.
2. **While** (*not converged or computational budget is not exhausted*)
3. Evaluate fitness of all feature subsets in the population based on $J(S_c)$.
4. Select the elite chromosome c_b to undergo Markov blanket based memetic operation.
5. Replace c_b with an improved new chromosome c'_b using Lamarckian learning.
6. Perform evolutionary operators based on restrictive selection, crossover, and mutation.
7. **End While**

END

Figure 4.1: Markov blanket embedded genetic algorithm (MBEGA) for gene selection

At the start of the MBEGA search, an initial GA population is initialized ran-

domly with each chromosome encoding a candidate feature subset. In the present work, each chromosome is composed of a bit string of length equal to the total number of features in the feature selection problem of interest. Using binary encoding, a bit of '1' ('0') implies the corresponding feature is selected (excluded). The fitness of each chromosome is then obtained using the objective function, $Fitness(c) = J(S_c)$ (For more details, the reader is referred to Section 3.2.2).

In each GA generation, the elite chromosome, i.e., the one with the best fitness value then undergoes Markov blanket based memetic operators/local search in the spirit of Lamarckian learning [4, 7]. The Lamarckian learning forces the genotype to reflect the result of improvement through placing the locally improved individual back into the population to compete for reproductive opportunities. Two memetic operators, namely an *Add* operator that inserts a feature into the elite chromosome, and a *Del* operator that removes some of the existing features from the elite chromosome, are introduced in the MBEGA. The important question is which feature to add and which feature to delete. Ideally, each deleted feature should be covered by another selected feature in the existing GA solution. This requirement is fast fulfilled by using the Markov blanket concept.

We illustrate our method here. For a given selected subset encoded in the chromosome c , we define $\mathbf{X}$ and $\mathbf{Y}$ as the sets of selected and excluded features encoded in c , respectively. The purpose of the *Add* operator is to select a highly correlated feature Y_i using the *C-correlation* measure from $\mathbf{Y}$ and then move it to $\mathbf{X}$. The *Del* operator on the other hand selects a highly correlated feature X_i from $\mathbf{X}$ and re-

moves other features that are covered by X_i using the approximate Markov blanket. If there is no feature in the approximate Markov blanket of X_i , the operator also tries to delete X_i itself. The details of these two memetic operators are outlined in Figure 4.2 and Figure 4.3.

Add Operator:
BEGIN

1. Rank the features in $\mathbf{Y}$ in a descending order based on *C-correlation* value.
2. Select a feature Y_i in $\mathbf{Y}$ using linear ranking selection [84] so that the larger the *C-correlation* of a feature in $\mathbf{Y}$ the more likely it will be selected.
3. Add Y_i to $\mathbf{X}$.

END

Figure 4.2: *Add* operation

Del Operator:
BEGIN

1. Rank the features in $\mathbf{X}$ in a descending order based on *C-correlation* value.
2. Select a feature X_i in $\mathbf{X}$ using linear ranking selection [84] so that the larger the *C-correlation* of a feature in $\mathbf{X}$ the more likely it will be selected.
3. Eliminate all features in $\mathbf{X} - \{X_i\}$ which are in the approximate Markov blanket of X_i . If no feature is eliminated, remove X_i itself.

END

Figure 4.3: *Del* operation

It is worth noting that the *C-correlation* measure of each feature (i.e., action (1) in Figure 5.1 and Figure 5.2) needs to be calculated only once. This feature ranking information is then archived for use in the *Add* and *Del* operations, for fine-tuning the GA solutions throughout the entire search. We further illustrate the details of

the *Add* and *Del* operations in Figure 4.4. For instance, $F5$ and $F4$ represent the highest and lowest ranked features in $\mathbf{Y}$, while $F3$ and $F6$ are the highest and lowest ranked features in $\mathbf{X}$, respectively. Further, $F3$ is the approximate Markov blanket of $F1$ and $F6$. In the *Add* operation, $F5$ is the most likely feature to be moved to $\mathbf{X}$. While in the *Del* operation, $F1$ and $F6$ will be deleted from $\mathbf{X}$ since they are already blanketed by $F3$. The two most probable resultant chromosomes after undergoing the *Add* and *Del* operations are also depicted in Figure 4.4.

Figure 4.4: Markov blanket based memetic operations (AMB denotes Approximate Markov Blanket)

It is possible to quantify the computational complexity of the two memetic operators based on the search range l , which defines the maximum numbers of the *Add* and *Del* operations. Therefore, with l possible *Add* operations and l possible *Del* operations, there are a total of l^2 possible combinations of *Add* and *Del* operations applied on a chromosome. In this study, we use an improvement first strategy (described in Section 3.2.3) to apply these l^2 combinations, since it has been shown to be more efficient than other strategies in Chapter 3.

After applying the above Lamarckian learning process on the elite chromosome, the GA population then undergoes the usual evolutionary operations including linear ranking selection [84], uniform crossover, and mutation operators with elitism [12]. However, if prior knowledge on the optimum number of features is available, we permit the incorporation of such information in our proposed memetic algorithm by constraining the number of bits '1' in each chromosome to a maximum of m in the evolutionary search process. To do so, we proposed using restrictive crossover and mutation (described in Figures 3.6 and 3.8, respectively) instead of simple GA operators, such that the number of bits '1' in each chromosome does not violate the constraint posed by the prior knowledge on m during the evolutionary search.

4.3 Empirical Study

In this section, we study the performance of the proposed MBEGA method against recent filter and wrapper feature selection algorithms using both high-dimensional synthetic problems and real-world microarray data. In particular, we consider the FCBF [28,90], BIRS [94] and standard GA feature selection algorithms. These algorithms have previously been successfully used for gene selection and demonstrated to attain promising performance [13,14,90,94]. Here, we provide some brief overviews of these algorithms.

FCBF (Fast Correlation-Based Filter), proposed by Yu and Liu [28,90], is a fast correlation-based filter method. It begins by selecting a subset of relevant features with *C-correlation* values that satisfy some pre-defined threshold γ . These relevant

features are then sorted in a descending order based on their *C-correlation* values. Using the sorted feature list, redundant features are eliminated one-by-one in a descending order. A feature is redundant only if it has an approximate Markov blanket. The obtained feature subset thus contains only predominant features and zero redundant features.

BIRS (Best Incremental Ranked Subset) [94] uses a similar scheme as the FCBF but the goodness of features is determined based on the classifier employed. Like FCBF, BIRS begins by ranking the genes according to some measure of interest and then sequentially selects the ranked features one-by-one based on their incremental usefulness. Given the current selected feature subset and the learning algorithm, a feature is incrementally useful if its addition to the selected feature subset results in statistically significant improvement in classification accuracy. Hence, after the initial ranking, BIRS makes calls to the classifier as many times as the number of features. Based on the ranking measure used in the first step, BIRS is categorized either as BIRS_F or BIRS_W , which rank features based on the *C-correlation* measure (i.e. symmetrical uncertainty between feature F_i and the class C) or individual predictive power, respectively. In [94], both BIRS_W and BIRS_F have been shown to converge to similar feature size at competitive classification accuracy but the latter completing the task at lesser time effort. Hence, we consider BIRS_F as the representative method in the present study. Note that the use of BIRS_F also provides a fair comparison with FCBF and MBEGA, since all of these methods use the same feature ranking measure.

GA feature selection is similar to MBEGA except that it does not involve any memetic operators. Both GA and MBEGA use the same parameter setting of population size = 50, crossover probability = 0.6, and mutation rate = 0.1. MBEGA or GA search stops when convergence to the global optimal has occurred or the maximum computational budget allowable (i.e., 2000 fitness functional calls) is reached. It is worth noting that the fitness function calls made to $J(S_c)$ in the memetic operations are also included as part of the total fitness function calls. The memetic operation range l in MBEGA is set to 4. These configurations are unchanged in our study on both the synthetic and microarray data.

FCBF, BIRS, and the classifiers (C4.5 and SVM) used in the following studies have been developed using the Weka environment [95]. The parameters for FCBF, C4.5, and SVM were based on the defaults in Weka. For BIRS, the configurations reported in [94] are used.

4.3.1 Empirical Results on Synthetic Data

We first present the results of our study on four synthetic datasets. We take an often used dataset Corral [96] and extend it with additional redundant and irrelevant features. Our purpose is to evaluate the four algorithms on different combinations of redundant and irrelevant features. The algorithms are expected to perform well on some but not all combinations.

The first dataset is the Corral data [96], which consist of 6 boolean features ($A0, A1, B0, B1, I, R75$) and a boolean class C defined by $C = (A0 \wedge A1) \vee (B0 \wedge$

$B1$). The features $A0, A1, B0$ and $B1$ are independent to each other, feature I is uniformly random and irrelevant to class C . $R75$ matching 75% of C is redundant. The second dataset, Corral-46 generated in [28], is obtained by introducing more irrelevant and redundant features to the original Corral data. It includes the optimal feature subset $(A0, A1, B0, B1)$, 14 irrelevant features, and 28 additional redundant features. Among the 28 additional redundant features, $A0, A1, B0$ and $B1$, each has 7 redundant features that match at levels of $1, 15/16, 14/16, \dots, 10/16$. The third dataset, Corral-50, is created by adding to Corral-46 with redundant features $R75, R80, R85$ and $R90$, that match C at the level of 75%, 80%, 85% and 90%, respectively. In the following text, we denote the subset of $R75, R80, R85$, and $R90$ as $R+$ for the sake of brevity. All features $R+$ are all highly correlated to C and have larger C -correlation measure than features in the optimal feature subset. To test the methods on problems of high-dimensions and high-redundancies like the case of microarray data, we generate the last dataset Corral-10000 from Corral-50 with additional 9964 irrelevant features.

The four synthetic datasets considered here are also summarized in Table 4.1. In the present study, we consider C4.5 as the classifier for feature subset evaluation since it provides the optimal subset of $(A0, A1, B0, B1)$ at a theoretical prediction accuracy of 100%.

The feature selection performances of FCBF, BIRS, GA and MBEGA on the synthetic datasets using ten 10-fold cross-validations with C4.5 classifier are reported in Table 4.2. In particular, the selected feature subset, number of selected features,

Table 4.1: Summary of the four synthetic datasets

Dataset	Features
Corral	OS: $A0, A1, B0, B1$ RF: $R75$ IF: I
Corral-46	OS: $A0, A1, B0, B1$ RF: $A0_{\{1,15/16,\dots,10/16\}}, A1_{\{1,15/16,\dots,10/16\}}, B0_{\{1,15/16,\dots,10/16\}}, B1_{\{1,15/16,\dots,10/16\}}$ IF: $I0, I1, \dots, I13$
Corral-50	OS: $A0, A1, B0, B1$ RF: $A0_{\{1,15/16,\dots,10/16\}}, A1_{\{1,15/16,\dots,10/16\}}, B0_{\{1,15/16,\dots,10/16\}}, B1_{\{1,15/16,\dots,10/16\}}$ $R90, R85, R80, R75$ IF: $I0, I1, \dots, I13$
Corral-10000	OS: $A0, A1, B0, B1$ RF: $A0_{\{1,15/16,\dots,10/16\}}, A1_{\{1,15/16,\dots,10/16\}}, B0_{\{1,15/16,\dots,10/16\}}, B1_{\{1,15/16,\dots,10/16\}}$ $R90, R85, R80, R75$ IF: $I0, I1, \dots, I9963$

OS: optimal subset; **RF:** redundant features; **IF:** irrelevant features.

$A0_{\{1,15/16,\dots,10/16\}}$ denotes the subset of 7 features matching $A0$ at levels of 1, 15/16, 14/16, ..., 10/16. (Similar definition applied to $A1_{\{1,15/16,\dots,10/16\}}, B0_{\{1,15/16,\dots,10/16\}}, B1_{\{1,15/16,\dots,10/16\}}$)

Table 4.2: Feature selection by each algorithm on synthetic data

		FCBF	BIRS	MBEGA	GA
Corral	S_c	(R75,A0,A1,B0,B1)	(R75)	(A0,A1,B0,B1)	(A0,A1,B0,B1)
	$ S_c $	5	1	4 ± 0	4 ± 0
	acc	96.02	75.00	100.00 ± 0	100.00 ± 0
Corral-46	S_c	(A0,A1,B0,B1)	(A0,A1,B0,B1)	(A0,A1,B0,B1)	(...)
	$ S_c $	4	4	4 ± 0	12.4 ± 0.4
	acc	100.00	100.00	100.00 ± 0	100.00 ± 0
Corral-50	S_c	(R+,A0,A1,B0,B1)	(R90)	(A0,A1,B0,B1)	(...)
	$ S_c $	8	1	4 ± 0	16.3 ± 1.2
	acc	97.97	90.63	100.00 ± 0	100.00 ± 0
Corral-10000	S_c	(R+,A0,A1,B0,B1)	(R90)	(A0,A1,B0,B1)	(...)
	$ S_c $	8	1	4 ± 0	13.6 ± 2.5
	acc	97.97	90.63	100.00 ± 0	82.51 ± 4.72

S_c : selected feature subset; $|S_c|$: average number of selected features; acc : classification accuracy.

and average classification accuracy are tabulated. Due to the stochastic nature of MBEGA and GA, the average feature selection results of MBEGA and GA for ten independent runs are reported. The maximum number of selected features in each chromosome, m , is set to 50.

The results in Table 4.2 show that FCBF, BIRS, and MBEGA locate the optimal feature subset at a classification accuracy of 100% on the Corral-46 dataset. GA fails to find the optimal subset on the Corral-46 dataset, but it performs better

than FCBF and BIRS on Corral dataset. Only MBEGA successfully identifies the optimal feature subset at perfect accuracy of 100% on all the four synthetic datasets. FCBF manages to identify $A0, A1, B0, B1$ but fails to eliminate the features in $R+$. Since all features in $R+$ have higher *C-correlation* values than any feature in the optimal feature subset. FCBF fails to identify these redundant features based on the approximate Markov blanket. On the other hand, BIRS is a sequential forward search method that is incapable of considering interactions between features, hence it is more likely to find suboptimal results. BIRS selects the feature with the highest *C-correlation* value. For instance, on dataset Corral-50, BIRS selects only the single top ranked feature, $R90$, with a classification accuracy of 90.63% and any additional features do not bring any improvement. GA selects much more features than the other methods. Without Markov blanket based memetic operators, GA alone is inefficient in eliminating redundant features under the limited computational budget.

4.3.2 Experimental Results on Microarray Data

In this section, we consider some real-world microarray datasets having a significantly large number of features (genes). In particular, 11 publicly available datasets in [87,97] are considered in the present study. A brief overview of these 11 datasets is summarized in Table 4.3 (for more details about these datasets, the reader is referred to Appendix D).

In many earlier work, researchers typically split the original dataset into two sets, a training set and a test set in a random fashion. Gene selection is then

Table 4.3: Description of 11 Microarray Datasets

Dataset	#Total Genes (T)	#Instances (n)	#Classes
Colon Tumor	2000	62	2
CNS*	7129	60	2
ALL-AML	7129	72	2
Breast Cancer	24481	97	2
Lung Cancer	12533	181	2
Ovarian Cancer	15154	253	2
ALL-AML-3	7129	72	3
ALL-AML-4	7129	72	4
Lymphoma	4026	62	3
MLL	12582	72	3
SRBCT	2308	83	4

*CNS: Central Nervous System

performed on the training set and the goodness of selected genes is assessed using the unseen test set. However, due to the small number of instances, such an approach is now recognized by the community as unreliable. Instead, Ambroise and McLachlan [74] suggested splitting the data using external¹ (10-fold) cross validation or .632 bootstrap. Furthermore, a comparison of various error estimation methods on microarray classification [69] also suggests that .632 bootstrap may be more appropriate than other estimators including re-substitution estimator, k-fold cross-validation, and leave-one-out estimation. Taking this cue, we employed a balanced external .632 bootstrap to evaluate the performances of the feature selection algorithms considered in this study. The .632 bootstrap involves sampling a training set with replacement from the original dataset. The test set is formed by those samples omitted from the training set. The .632 bootstrap is repeated K times and

¹For more discussion on internal and external validation, the reader is referred to Section 2.6.4.

the final bootstrap error estimator $\epsilon_{b.632}$ is defined as:

$$\epsilon_{b.632} = \frac{1}{K} \sum_{i=1}^K (0.368\alpha_i + 0.632\beta_i) \quad (4.2)$$

where α_i and β_i are the training error and test error on the i^{th} re-sampling, respectively. Following the work in [74], the bootstrap samples are formed with $K = 30$ replicates. Each instance in the original dataset is made to appear exactly 30 times in the balanced bootstrap training samples. Feature selection is then performed using only the training samples. In BIRS, MBEGA and GA, a cross validation scheme is applied on the training samples for feature subset evaluation. Finally, the test error is estimated on the unseen test samples. The classification accuracy is then estimated using Equation (4.2).

In the present study, the support vector machine (SVM) is chosen as the classifier since it has been demonstrated in the literature to outperform many existing machine learning methods (e.g., KNN, C4.5, Naive Bayes, etc) on two-class [92] or multi-class [76, 87] microarray classification problems. For multiclass datasets, a One-Versus-Rest strategy is applied, since it has shown superior performance on microarray classification to other strategies in [76]. Further, since the linear kernel SVM has been shown to generate good performance sufficient for microarray classification [88, 98], we consider it in the present study. The penalty parameter of SVM is chosen using a method similar to that in [76], which empirically optimizes the penalty parameter over the set of values $\{0.001, 0.01, 0.1, 1, 10, 100\}$. A value of 1 is selected for all the datasets except for Colon Tumor and Central Nervous System, for which

the optimal value is 0.1.

In MBEGA and GA, the SVM radius margin bound [79] is used to estimate the generalization error $J(S_c)$ with only a single run of the training dataset. For BIRS, SVM with 5-fold cross validation is used for estimating the classification accuracy, i.e., the evaluation procedure follows that described in [94] to avoid changes to the original algorithm proposed.

As previous studies have considered no more than 50 and 150 genes as optimal feature subset on two-class datasets [18, 19, 74, 85, 88–92] and multi-class datasets [16, 17, 87], respectively, we constrained the maximum number of selected features m in each chromosome to these bounds in both MBEGA and GA. Because the *C-correlation* measure is not designed for continuous data, we use Fayyad and Irani’s Minimum Description Length (MDL) method [99] to discretize the microarray data as a pre-processing step.

Table 4.4 presents the average classification accuracy, average number of selected genes, and average running time for each feature selection algorithm on the eleven datasets over 30 runs of .632 bootstraps.

Classification Accuracy

In Table 4.4, the best average classification accuracy among all four algorithms on each dataset is highlighted in bold typeface. Overall, MBEGA, FCBF and GA are observed to produce the best accuracy on 6/11, 4/11 and 1/11 datasets, respectively. Statistical test using random permutation test [76] at significance level of 0.05 shows that MBEGA outperforms GA in terms of accuracy on all datasets except the Lym-

Table 4.4: Performance of feature selection algorithms on microarray datasets

		FCBF	BIRS	MBEGA	GA
Colon Tumor (2000 $\times$ 62)	<i>acc</i>	84.54	76.49	85.66	81.01
	$\overline{ S_c }$	21.0	1.8	24.5	23.3
	t(s)	1.1	56.0	70.6	66.8
CNS (7129 $\times$ 60)	<i>acc</i>	73.29	64.14	72.21	68.30
	$\overline{ S_c }$	52.8	1.7	20.5	24.1
	t(s)	4.2	36.6	81.1	78.5
ALL-AML (7129 $\times$ 72)	<i>acc</i>	93.82	89.34	95.89	92.31
	$\overline{ S_c }$	32.8	2.0	12.8	25.2
	t(s)	5.6	378.2	112.3	118.0
Breast Cancer (24481 $\times$ 97)	<i>acc</i>	75.89	67.98	80.74	68.81
	$\overline{ S_c }$	127.9	2.3	14.5	22.1
	t(s)	28.3	3766.4	266.9	243.8
Lung Cancer (12533 $\times$ 181)	<i>acc</i>	95.70	97.31	98.96	98.06
	$\overline{ S_c }$	66.3	2.5	14.1	24.4
	t(s)	30.8	1273.1	1041.7	1025.5
Ovarian Cancer (15154 $\times$ 253)	<i>acc</i>	99.91	98.50	99.71	99.43
	$\overline{ S_c }$	31.3	2.0	9.0	23.3
	t(s)	26.7	2098.9	2689.5	2588.5
ALL-AML-3 (7129 $\times$ 72)	<i>acc</i>	95.76	84.49	96.64	90.96
	$\overline{ S_c }$	77.7	2.4	18.1	75.1
	t(s)	3.7	1175.4	176.6	174.2
ALL-AML-4 (7129 $\times$ 72)	<i>acc</i>	92.38	77.60	91.93	88.06
	$\overline{ S_c }$	99.3	2.2	26.2	74.4
	t(s)	4.4	1329.8	234.3	218.4
Lymphoma (4026 $\times$ 62)	<i>acc</i>	94.85	86.11	97.68	98.99
	$\overline{ S_c }$	48.5	2.0	34.3	74.9
	t(s)	2.1	287.8	142.6	139.1
MLL (12582 $\times$ 72)	<i>acc</i>	95.64	84.00	94.33	92.22
	$\overline{ S_c }$	101.6	3.7	32.1	75.4
	t(s)	8.4	2593.8	182.1	165.0
SRBCT (2308 $\times$ 83)	<i>acc</i>	98.94	86.66	99.23	95.77
	$\overline{ S_c }$	98.6	4.1	60.7	78.4
	t(s)	2.2	386.8	246.2	232.6

acc: average classification accuracy; $\overline{|S_c|}$: average number of selected genes; t(s): running time (second).

phoma dataset. The accuracies of FCBF and MBEGA are not significantly different from each other on 8/11 datasets. Out of the remaining 3 datasets (Breast Cancer, Lung Cancer, and Ovarian Cancer), MBEGA performs better than FCBF on the Breast Cancer and Lung Cancer datasets, while poorer on the Ovarian Cancer dataset. BIRS is observed to have the lowest classification accuracy among the four methods on all 11 datasets.

Number of Selected Genes

With respect to the number of selected genes, BIRS selects the smallest feature subset with no more than 4 genes on all datasets but it does so at the expense of poor classification accuracy. On the other hand, MBEGA selects a smaller feature subset at a classification accuracy that is competitive or superior to both FCBF and GA. Since FCBF selects predominant genes based on the correlation measure and ignores the inductive algorithm, some of the selected genes are actually redundant as they do not contribute to any improvement in the classification accuracy. In contrast, GA identifies suitable gene subsets solely based on the predictive ability of the inductive algorithm. MBEGA on the other hand takes into account of both correlation redundancy among genes as well as their cooperative classification ability via a wrapper scheme. Consequently it is able to eliminate more redundant features without deteriorating the classification accuracy and select genes more suitable for the inductive algorithm. The results in Table 4.4 suggest that on most of the datasets, MBEGA obtains competitive or superior classification accuracy to FCBF using only one-eighth to one-half the number of genes selected by FCBF. MBEGA also selects

much fewer genes than GA but with significantly better accuracy. Especially on the multi-class datasets, MBEGA selects only about one-third the number of genes selected by GA. We also note that FCBF has a larger standard deviation in the number of selected features than MBEGA, particularly for ALL-AML, Breast Cancer, Lung Cancer, ALL-AML-4, Lymphoma, and MLL datasets, which suggests that MBEGA is more robust on different bootstrap re-sampling datasets than FCBF. The robustness of the various methods is further investigated later in Section 4.3.2.

Average Running Time

The average running time of the algorithms are also reported in Table 4.4. The Pearson’s correlation coefficient between the running time and other external factors (i.e., number of instances and number of classes, and total number of genes) are then summarized in Table 4.5.

Table 4.5: Correlation coefficient between running time and other external factors

Correlation coefficient	FCBF	BIRS	MBEGA	GA
t(s) vs. #Instances	0.792	0.351	0.969	0.970
t(s) vs. #Classes	-0.500	-0.106	-0.296	-0.300
t(s) vs. Total # Genes	0.839	0.916	0.384	0.380
t(s): running time (second).				

From table 4.4, we can see that the running time of the filter method FCBF are lesser than 31 seconds on all the datasets, which are slightly correlated with the number of instances and the total number of genes as shown in Table 4.5. FCBF is much faster than all other algorithms as it evaluates features without involving the time consuming classification. For the three wrapper methods, most of the time

is spent on classification (i.e., fitness function call), hence other overheads (e.g., the ranking of *C-correlation* in BIRS and MBEGA, the evolutionary operations in MBEGA and GA) are considered to be negligible. The overall running time of BIRS, MBEGA or GA is a function of the total classification calls and the time taken to perform a single classification, which are dependent on external factors including the number of instances, number of classes, and total number of genes. The running times of MBEGA and GA are highly correlated to the number of instances but do not depend on the total number of genes. This makes MBEGA more suitable for microarray data which are characterized with thousands of genes but only have tens or hundreds of instances. MBEGA and GA employ identical classifier and maximum number of classification calls allowable. Hence, they incur similar running time on all the datasets. BIRS makes a single classification for each gene, therefore its running time is a function of the total number of genes. We validate this further in Table 4.5 where a high correlation coefficient of 0.916 is shown. The results in Table 4.4 also show that BIRS incurs approximately the same amount of running time as MBEGA and GA on five out of the six two-class datasets but when the number of genes increases tremendously, e.g., the Breast Cancer dataset has 24481 genes, the running time for BIRS shoots up to 3766.4 seconds which is 14 times longer than that of MBEGA (266.9 seconds). BIRS also takes more time than MBEGA and GA on all the multi-class datasets.

Robustness

In the microarray datasets, the number of instances available for learning is typically small. As such the gene selection results obtained in the bootstrap re-sampling process are likely to have large variances. This is generally unacceptable to biologists, who believe that a good algorithm should be one that identifies the crucial genes for the purpose of diagnosis, therapeutics, or prognosis of cancer consistently and not by chance. For this reason, here we also investigate the robustness of different gene selection algorithms.

In this study, we determine the robustness using Z-score analysis [14,15]. Z-score indicates the significance of the frequency a gene getting selected. A gene with a high Z-score suggests that it is not selected by chance and an algorithm that identifies a gene subset containing genes of higher Z-score is deemed to be more robust. The Z-score of gene i is defined as:

$$Z = \frac{S_i - E(S_i)}{\sigma_i} \quad (4.3)$$

where S_i is the frequency that gene i is selected, $E(S_i)$ is the expected number of times gene i is selected, while σ_i is the standard deviation for S_i . Let $\overline{|S_c|}$ be the average number of selected genes and T be the total number of genes, then the probability of gene i being selected in each run of bootstrap is denoted as, $P(S_i) = \overline{|S_c|}/T$. Whereupon, the expected number of times gene i being selected among the K runs of bootstrap is calculated as $E(S_i) = P(S_i) \cdot K$, and the standard deviation of S_i is derived from $\sigma_i = \sqrt{P(S_i) \cdot (1 - P(S_i)) \cdot K}$.

We sort the genes in a descending order in terms of their frequency of selection. The Z-scores of the top 50 selected genes for all the four algorithms on the 11 microarray datasets are plotted in Figure 4.5 and Figure 4.6. The x -axis denotes the sorted top 50 most selected genes and the y -axis denotes the Z-score. The results indicate that the top 50 genes selected by MBEGA have larger Z-scores than that of the other counterpart algorithms, which suggests MBEGA is more robust than the other algorithms.

Overall Performance

From the empirical results obtained so far, it is worth noting that each method has its strengths and limitations. In particular, FCBF obtains good classification accuracy in the least amount of running time but at the expense of selecting many more genes; BIRS selects the least number of genes but suffers in terms of classification accuracy and also requires a larger running time than the other counterparts; MBEGA attains the best accuracy and robustness in a reasonable time. For different purposes, different methods could be selected. However in practice, a method with better overall performance on all the evaluation criteria is favorable. To quantitatively investigate the overall performance, a rank (i.e., 1, 2, 3, 4) of each evaluation criterion is assigned to each algorithm on a given dataset. If there is no significant difference between multiple algorithms on a given criterion, an average rank is distributed among them (e.g., on Colon Dataset, the classification accuracy is not significantly different for FCBF and MBEGA, hence a rank of 1.5 is assigned to both). The summations of ranks on all the 11 datasets are tabulated in Table 4.6. The best

Figure 4.5: The Z-score of the top 50 most frequently selected genes (a) Colon Cancer, (b) Central Nervous System, (c) ALL-AML, (d) Breast Cancer, (e) Lung Cancer, (f) Ovarian Cancer.

Figure 4.6: The Z-score of the top 50 most frequently selected genes (a) ALL-AML-3, (b) ALL-AML-4, (c) Lymphoma, (d) MLL, (e) SRBCT.

performing algorithm on each criterion is highlighted in boldface. Summations of ranks for all the four evaluation criteria (shown in the last row of Table 4.6) suggest that MBEGA has a best compromise among all the four criteria.

Table 4.6: Summation ranks of four evaluation criteria

	FCBF	BIRS	MBEGA	GA
acc	19.5	41	18.5	31
$ S_c $	41.5	11	23.5	34
$t(s)$	11	38	30.5	30.5
Robustness	23	32	11	44
Sum	95	122	83.5	139.5

Comparison with Other Methods in the Literature

Since MBEGA uses GA and SVM, we also compare it with other GA or SVM based gene selection methods. We compared the most related work in the literature for the same datasets.

Firstly the GA based methods are considered. In [13, 14], Li and his colleagues proposed a GA/KNN method. This method is repeatedly applied on a training dataset for 20,000 to 40,000 times. Among the different selected gene subsets, the most frequently selected genes are used for classification on a hold-out test dataset. In [13], GA/KNN is reported to obtain a classification accuracy of 97.06% using the top 50 genes on the ALL-AML dataset. On the same dataset, MBEGA attains a competitive accuracy of 95.89% with only 12.8 genes. Furthermore, the .632 bootstrap validation scheme applied in this study is more reliable than the hold-out scheme.

The other two GA based methods, GA/MLHD and GA/SVM, are proposed in [16] and [17], respectively. Both of them are applied to the same multi-class NCI60 dataset with 1000 pre-selected genes and the results are compared in [17]. GA/MLHD is observed to attain a leave-one-out cross validation (LOOCV) accuracy of 70.73% using 12 selected genes. GA/SVM is reported to outperform GA/MLHD with a LOOCV accuracy of 88.52% using 40 genes. In this study, we compare MBEGA with the superior method GA/SVM. It is worth noting that the accuracies reported in [17] are actually not evaluated on a test dataset which must be independent to the gene selection procedure. In particular, the gene selection is conducted using the whole NCI60 dataset as a training data and LOOCV accuracy on the same data is reported as the final result. This so called internal cross validation [73, 74] has been demonstrated to suffer from selection bias [74] or overfitting [73], especially on datasets of small sample size. To alleviate this problem, an external bootstrap method as described earlier has been suggested for gene selection in [69, 74]. Hence, to make a fair comparison, 30 runs of external .632 bootstraps are applied for both MBEGA and GA/SVM on the NCI60 dataset. The results show that MBEGA attains a better accuracy of 70.38% with 47.6 selected genes, while GA/SVM obtains an accuracy of only 65.79% with 40 selected genes.

We further compare MBEGA with SVM-based methods. In [88], Guyon and his colleagues proposed a gene selection method utilizing SVM and recursive feature elimination (RFE). Under an internal cross validation scheme, the SVM-RFE method selects 8 genes and attains an accuracy of 100% on ALL-AML data. On

Colon data, it obtains an accuracy of 100% with 16 genes. These results have been demonstrated to suffer from selection bias in [74]. The bias corrected results using external .632 bootstrap in [74] show that the best accuracy of SVM-RFE on ALL-AML and Colon data are lower than 95% and 85% respectively, although it uses more than 64 genes. Zhou and Mao [85] proposed another SVM-based gene selection method using LS bound measure and sequential forward selection (SFS). Using .632+ external bootstrap the LS bound-SFS method obtains a best accuracy of 97.68% with 50 selected genes on ALL-AML dataset and a best accuracy of 84.95% with 46 selected genes on Colon dataset. For comparison, MBEGA reaches an accuracy of 95.89% on ALL-AML data and 85.66% on Colon data, which are better than that of SVM-RFE and competitive with that of LS bound-SFS. Interestingly, MBEGA achieves this with much less genes (i.e., 12.8 for ALL-AML and 24.5 for Colon).

4.4 Summary

In this chapter, we have proposed a novel Markov blanket embedded Genetic Algorithm (MBEGA) for gene selection. We also take representative methods from each of filter (FCBF), wrapper (BIRS), and standard GA to study their strengths and weaknesses in the present work. Empirical studies on 4 synthetic and 11 microarray datasets suggest that FCBF obtains good classification accuracy with least amount of running time but selecting many more features; BIRS selects the least number of features but suffers in terms of classification accuracy and running time; MBEGA

attains best accuracy and robustness in a reasonable time, it also achieves the best compromise on all the four evaluation criteria, i.e., classification accuracy, number of selected genes, computational cost, and robustness. Further comparison study to other feature selection methods in the literature also suggests MBEGA generates competitive or superior performances. MBEGA is able to eliminate irrelevant and redundant features efficiently using the Markov blanket and predictive power of the wrapper model, thus leading to the identification of a smaller set of reliable genes for biologists to concentrate on in their wet-lab study. Furthermore, it is worth noting that together with text mining on biological literature [17] or tools such as RSVP [100], MBEGA serves to enhance the process of candidate biomarkers discovering as well as assist researchers in analyzing the increasing biological research data.

Chapter 5

Comparing Memetic Algorithms For Gene Selection

In the previous two chapters, we have proposed two MAs, namely WFFSA and MBEGA, for gene selection. Both are synergies of the Genetic Algorithm (wrapper methods) and local search methods (based on filter methods) under a memetic framework. In particular, WFFSA fine-tunes the population of GA solutions by adding or deleting features based on univariate feature filter ranking methods. On the other hand, MBEGA fine-tunes the GA population by adding the relevant features, removing the redundant and irrelevant features using the Markov blanket approach.

We note that both WFFSA and MBEGA add or delete features based on filter methods which are totally independent of the GA fitness function. In other words, these filter methods use a different set of criteria that do not include fitness values for defining the qualities of the genes or features.

In this chapter, we present a comparison study on three variants of feature selection MAs using synthetic and real microarray data. Besides WFFSA and MBEGA,

a sensitivity analysis based MA (SABMA) that fine-tunes the GA population based on the sensitivity of the fitness function is also considered.

The rest of this chapter is organized as follows. Section 5.1 presents SABMA while Section 5.2 presents the numerical results obtained from our empirical study on synthetic and real microarray datasets. Analysis of the numerical results and some discussions are also presented in the section. Finally, Section 5.3 concludes this chapter.

5.1 Sensitivity Analysis Based MA

Sensitivity Analysis Based MA (SABMA) uses a similar memetic framework as the WFFSA and MBEGA, i.e., a synergy of GA and LS. In contrast, SABMA employs a different LS method which adds and deletes features on GA solutions based on sensitivity analysis of the GA fitness function.

Several authors have suggested using the change in the fitness function when one feature is removed or added as a criterion to evaluate the importance of that feature [11, 88]. Such a scheme is commonly known as a sensitivity analysis. A greater change in the fitness value when a feature is added or removed suggests that the feature has a higher degree of importance. In this study, the fitness function considered $J(S_c)$ is based on the radius margin bound of SVM.

$$J(S_c) = \frac{1}{d} R^2 \|W\|^2 \quad (5.1)$$

where d is the number of training samples, and the radius R and margin W are calculated as follows:

$$R^2 = \max_{\beta} \sum_{i=1}^l \beta_i K(\mathcal{X}_i, \mathcal{X}_j) - \sum_{i,j=1}^l \beta_i \beta_j K(\mathcal{X}_i, \mathcal{X}_j) \quad (5.2)$$

$$\|W\|^2 = 2 \sum_{i,j=1}^l \alpha_i^0 \alpha_j^0 \mathcal{Y}_i \mathcal{Y}_j K(\mathcal{X}_i, \mathcal{X}_j) \quad (5.3)$$

where $K(\mathcal{X}_i, \mathcal{X}_j)$ denotes the kernel function of training samples $\mathcal{X}_i$ and $\mathcal{X}_j$. α^0 is the optimal parameter vector of the SVM model after training.

Based on the definition of the radius margin bound, the sensitivity of a feature f on $J(S_c)$ is defined as:

$$Sensitivity(f) = \frac{1}{d} |R^2 \|W\|^2 - R_{(f)}^2 \|W_{(f)}\|^2| \quad (5.4)$$

where $R_{(f)}$ is the new radius value when feature f is added (if $f \in \mathbf{Y}$) or deleted (if $f \in \mathbf{X}$). $\|W_{(f)}\|$ denotes the new margin value when feature f is added (if $f \in \mathbf{Y}$) or deleted (if $f \in \mathbf{X}$). For the sake of simplicity and to reduce computational complexity, the model parameter vector α^0 is assumed to be relatively stable when a feature is added or deleted [101], hence avoiding the need to retrain the SVM model. The new radius and margin are then calculated using the following equations:

$$R_{(f)}^2 = \max_{\beta} \sum_{i=1}^l \beta_i K_{(f)}(\mathcal{X}_i, \mathcal{X}_j) - \sum_{i,j=1}^l \beta_i \beta_j K_{(f)}(\mathcal{X}_i, \mathcal{X}_j) \quad (5.5)$$

$$\|W_{(f)}\|^2 = 2 \sum_{i,j=1}^l \alpha_i^0 \alpha_j^0 \mathcal{Y}_i \mathcal{Y}_j K_{(f)}(\mathcal{X}_i, \mathcal{X}_j) \quad (5.6)$$

where $K_{(f)}$ denotes the new kernel function value after feature f is added or deleted. Since linear kernel function $K(\mathcal{X}_i, \mathcal{X}_j) = \mathcal{X}_i \cdot \mathcal{X}_j$ is considered in this study, $K_{(f)}$ could be easily calculated as $K_{(f)}(\mathcal{X}_i, \mathcal{X}_j) = K(\mathcal{X}_i, \mathcal{X}_j) \pm \mathcal{X}_{if} \cdot \mathcal{X}_{jf}$, where $\pm$ correspond to adding or deleting feature f , respectively. $\mathcal{X}_{if}$ and $\mathcal{X}_{jf}$ represent the values of training samples $\mathcal{X}_i$ and $\mathcal{X}_j$ on feature f , respectively. For more details of SVMs and the radius margin bound, the reader is referred to Appendix B.

Features in the selected subset $\mathbf{X}$ and excluded subset $\mathbf{Y}$ are then ranked based on their sensitivities using Equation 5.4. The *Add* and *Del* operators of SABMA are depicted in Figure 5.1 and 5.2, respectively.

Add Operator in SABMA:
BEGIN

1. Rank all features in $\mathbf{Y}$ based on $Sensitivity(f)$ in descending order.
2. Select a feature Y_i in $\mathbf{Y}$ using linear ranking selection [84] such that the higher the quality of a feature in $\mathbf{Y}$, the more likely it will be selected to move into $\mathbf{X}$.
3. Add Y_i to $\mathbf{X}$.

END

Figure 5.1: *Add* operator for SABMA

The *Add* and *Del* operators are applied on GA solutions using the improvement first strategy (described in Section 3.2.3).

***Del* Operator in SABMA:**
BEGIN

1. Rank all features in $\mathbf{X}$ based on $Sensitivity(f)$ in ascending order.
2. Select a feature X_i in $\mathbf{X}$ using linear ranking selection [84] such that the lower the quality of a feature in $\mathbf{X}$, the more likely it will be selected to move into $\mathbf{Y}$.
3. Remove X_i to $\mathbf{Y}$.

END

Figure 5.2: *Del* operator in SABMA

5.2 Comparison Study on WFFSA, MBEGA, and SABMA

In this section, we compare the performances of WFFSA, MBEGA, and SABMA, on both synthetic and real world microarray data. The standard GA (i.e., without local search) is then used as the baseline method for comparison. For fair comparison, all algorithms use the same GA parameter configurations with population size = 50, crossover probability = 0.6, and mutation rate = 0.1. The stopping criteria for all algorithms are defined by 1) convergence to the global optimal or 2) maximum computational budget of 2000 fitness functional calls is reached. It is worth noting that the fitness function calls made to $J(S_c)$ in the LS are also included as part of the total fitness function calls for fair comparison to the GA. The maximum number of bit '1' in the chromosome m is set to 50.

To evaluate the performance of each algorithm, we consider a balanced .632+ external bootstrap [69] for estimating the prediction accuracy of a given gene subset.

The external bootstrap is repeated 30 times for each dataset and the average results are reported. We compare each algorithm based on three performance criteria, i.e., classification accuracy, number of selected genes, and robustness, as used in Chapter 4. WFFSA, MBEGA, and SABMA are assumed to share a similar computational complexity of $O(pg + l^2wg/2)$. For more details, please refer to Section 3.2.5.

5.2.1 Synthetic Data

We begin our study of the proposed MAs on the synthetic datasets since the true optimal gene subset is known beforehand. All the three MAs successfully converge to the optimal feature subset in the synthetic Corral datasets which have been used in Chapter 4. Hence, these synthetic Corral datasets may not be suitable to investigate the strengths and weaknesses of all three MAs. As such, we include another two-class synthetic microarray data. It is composed of 4030 genes and 80 samples with 40 samples for each class label. The centroid of these two classes are located at $(-1,-1,-1)$ and $(1,1,1)$. Three groups of relevant genes, with each group containing 10 genes, are generated from a multivariate normal distribution using the following mean and covariance matrixes:

$$\mu = \begin{bmatrix} \mu_{-1} & \mu_{-1} & \mu_{-1} \\ \mu_1 & \mu_1 & \mu_1 \end{bmatrix} \quad \sigma = \begin{bmatrix} e & 0 & 0 \\ 0 & e & 0 \\ 0 & 0 & e \end{bmatrix}$$

where $\mu_{\mathbf{x}}$ is a 40×10 matrix with each element equal to x ; $\mathbf{0}$ is 10×10 matrix with each element equal to 0; $\mathbf{e}$ is a 10×10 symmetrical matrix with each diagonal element equal to 1 and the other elements equal to 0.9, i.e.,

$$\mathbf{e} = \begin{bmatrix} 1 & 0.9 & \dots & 0.9 \\ 0.9 & 1 & \dots & 0.9 \\ \vdots & \vdots & \ddots & \vdots \\ 0.9 & 0.9 & \dots & 1 \end{bmatrix}$$

The correlation between intra-group genes is 0.9, whereas the correlation between inter-group genes is 0. Hence, genes in the same group are redundant with each other and the optimal gene subset to separate the two classes consists of any 3 genes from different relevant gene groups. Another 4000 irrelevant genes are added to the data. Among these 4000 genes, 2000 are drawn from a normal distribution of $N(0,1)$ and the other 2000 genes are sampled with a uniform distribution of $U[-1,1]$.

Table 5.1: Feature selection by each algorithm on synthetic data

Algorithm	GA	WFFSA	SABMA	MBEGA
Accuracy	92.99	97.50	94.67	97.98
<i>#Genes</i>	34.1	35.5	17.7	9.7
<i>#Genes in Optimal Set</i>	2.6	3	2.7	3
<i>#Relevant Genes</i>	3.2	23.2	5.0	8.2
<i>#Redundant Genes</i>	0.6	20.2	2.3	5.2
<i>#Irrelevant Genes</i>	30.9	12.3	12.7	1.5
$\#Genes = \#Relevant\ Genes + \#Irrelevant\ Genes;$				
$\#Relevant\ Genes = \#Genes\ in\ Optimal\ Set + \#Redundant\ Genes.$				

The results of the feature selection by GA, WFFSA, MBEGA, and SABMA are tabulated in Table 5.1. First of all, the results in Table 5.1 suggest that classifica-

tion accuracy is improved when more genes within the optimal set and less irrelevant genes are selected, it is not sensitive on the number of selected redundant features. All the three MAs outperform GA in terms of classification accuracy. They consistently select more relevant genes, while GA fails to do so and converges to subsets that contain much more irrelevant genes at the end of the computational budget of 2000 fitness function calls. MBEGA selects the smallest subset of relevant genes with the use of Markov blanket for local learning. It also obtains the highest classification accuracy among all four methods. Since the filter ranking procedure in WFFSA considers individual gene contributions during local learning, it cannot identifying the interaction between genes. Therefore, unlike MBEGA, WFFSA is incapable of removing the redundant and irrelevant genes. While, considering the sensitivity of fitness function, SABMA is inefficient on identifying the optimal feature subset, it is also inefficient in eliminating irrelevant features.

5.2.2 Microarray Data

In this section, we consider several real-world microarray datasets having a significantly large number of features (genes). In particular, six datasets (i.e., Colon Cancer, Central Nervous System (CNS), ALL-AML, Breast Cancer, Lung Cancer, and Ovarian Cancer) in [97] are considered here. In Table 5.2, the average classification accuracy and average number of selected genes for each feature selection algorithm across the six datasets are reported.

The results in Table 5.2 are consistent with those reported in Section 5.2.1 on

Table 5.2: Average of classification accuracy and selected gene sized of feature selection algorithms, GA, WFFSA, SAMMA, and MBEGA, on microarray datasets

		GA	WFFSA	SABMA	MBEGA
Colon	acc	81.01	84.77	81.12	85.66
(2000 $\times$ 60)	$ S_c $	23.3	23.1	13.6	24.5
CNS	acc	68.30	72.30	68.86	72.21
(7129 $\times$ 60)	$ S_c $	24.1	22.1	11.1	20.5
ALL-AML	acc	92.31	96.81	93.57	95.89
(7129 $\times$ 72)	$ S_c $	25.2	29.6	16.7	12.8
Breast	acc	68.81	74.18	68.15	80.74
(24481 $\times$ 97)	$ S_c $	22.1	27.8	10.5	14.5
Lung	acc	98.07	99.12	98.72	98.96
(12533 $\times$ 181)	$ S_c $	24.4	28.6	15.1	14.1
Ovarian	acc	99.43	99.50	99.45	99.52
(15154 $\times$ 253)	$ S_c $	23.3	18.7	12.7	9.0

acc : classification accuracy; $|S_c|$: average number of selected genes.

the synthetic data. All MAs obtained a higher classification accuracy than the standard GA. This suggests all the LSs considered here to form the different MAs have been successful in fine-tuning the GA solutions effectively and efficiently. As a result, smaller subset of important genes that generates improved classification accuracies are found for all the datasets considered. Further, the results in Table 5.2 also suggest that WFFSA and MBEGA are competitive to each other in terms of classification accuracy. Both outperform each other on three out of the eleven datasets considered. Nevertheless, it is worth highlighting that once again MBEGA converges to more compact gene subsets than WFFSA by eliminating the redundant genes that exist. SABMA tends to obtain similar or better training accuracy than WFFSA and MBEGA, but it displays lower classification accuracies on unseen test data when compared to WFFSA and MBEGA, due to the effect of overfitting when

using a sensitivity based LS. Since SABMA employs a LS that considers the sensitivity of the fitness function, the feature subset found by SABMA is more likely to overfit on the training dataset hence perform poorly on the independent test dataset.

5.2.3 Robustness

We also evaluate the robustness of the three MAs and GA using Z-score [14, 15]. To do so, the selected genes have to be sorted according to their frequency of selection in a descending order. The Z-scores of the top 50 selected genes for all four algorithms on all microarray and synthetic datasets are plotted in Figure 5.3. The x -axis denotes the sorted top 50 most selected genes while the y -axis denotes the corresponding Z-score. The results obtained indicate that all three MAs are more robust than GA. The top 50 genes selected by WFFSA have slightly larger Z-scores than that of MBEGA, which suggests WFFSA is more robust than MBEGA. Both WFFSA and MBEGA are more robust than SABMA.

5.3 Summary

In this chapter, three MAs feature selection algorithms WFFSA, MBEGA, and SABMA are investigated and compared using synthetic and real world microarray data. With the inclusion of local learning *Add/Del* operators based on ReliefF ranking, Markov blanket, or sensitivity analysis in the standard GA, all MAs have been observed to converge to solutions with more compact feature/gene subsets at improved classification accuracy and robustness. In comparison, both WFFSA and

Figure 5.3: The Z-score of the top 50 most frequently selected genes on (a) Colon, (b) Central Nervous System, (c) ALL-AML, (d) Breast, (e) Lung, (f) Ovarian, (g) Synthetic.

MBEGA outperform SABMA in terms of classification accuracy and robustness, which suggests the choice of filter-based LS operator may be preferred over sensitivity based LS since the latter tends to overfit the training data. On the other hand, the prediction accuracy is not significantly different between WFFSA and MBEGA. While MBEGA generally finds more compact feature subset in final solution than WFFSA, WFFSA displays better robustness than MBEGA. Hence, both serve as suitable alternatives for identifying candidate biomarkers in the research of cancer and other diseases.

Chapter 6

Simultaneous Identification of Full Class Relevant and Partial Class Relevant Genes

In Chapters 4 and 5, we have shown that MBEGA is efficient for eliminating irrelevant and redundant genes on microarray data. However, it remains difficult for most existing feature selection methods including MBEGA to identify partial class relevant genes on multiclass problems. In this chapter, we propose a multiobjective memetic algorithm for identifying the crucial genes in multiclass problems.

6.1 Introduction and Motivation

Microarray technology has provided a promising tool for cancer diagnosis. Various statistical and machine learning methods have been successfully used for aiding in cancer identification and prediction. Most of these techniques were tailored for handling two-class classification problems. For example, signal-to-noise ratio [89], t-test [102], between-groups to within-groups ratio [91], support vector machine

(SVM) based on recursive feature elimination method [88], and least squares bound method [85], etc. are among those applicable only to two-class problems. Microarray data with multiple class labels remains to be a big challenge due to the significant statistical and analytical implications.

In recent years, multiclass problems on microarray data have drawn increasing attention in the field of bioinformatics. In particular, there have been increasing number of studies on identifying the crucial genes for discriminating multiple cancer types. Ramaswamy [103] et al. investigated the use of a binary SVM and binary feature selection methods using a One-Versus-All (OVA) classification schema for handling multiclass problems. Nguyen and Rocke [104] presented the use of partial least squares method for feature selection and classification of cancer samples from multiple classes. Ooi and Tan [16] proposed a GA/MLHD method for multiclass gene selection. It is a GA-based wrapper feature selection schema that employs maximum likelihood (MLHD) as the induction algorithm. A similar feature selection method based on GA and SVM is also investigated in Liu et al. [17]. Ooi et al. [105] also introduced another correlation-based filter gene selection method for multiclass problems, which takes into consideration the criteria of relevance, redundancy, and the degree of differential prioritization (DDP). Tibshirani et al. [106] proposed a ‘shrunk centroid’ algorithm for multiclass gene selection. While Yeung et al. [107] extended their work to take the dependency between genes and repeated measurements into considerations. They also investigated a Bayesian model averaging (BMA) approach [108] for gene selection on multiclass microarray data.

Comparative studies on multiclass gene selection problems can also be found in Li et al. [87] and Statnikov et al. [76], where the performances of various state-of-art feature selection methods and classification algorithms for solving multiclass microarray datasets were reported.

Multiclass cancer classification using microarray data has provided the feasibility of cancer diagnosis across all of the common malignancies in parallel. Using multiclass cancer feature selection approaches, it is now possible to identify genes that are relevant to a set of cancer types. However, besides identifying the relevant genes, it would be more useful for biologists to also possess the knowledge of which specific cancer or subset of cancer types a gene is relevant to. In this chapter, we introduce two new definitions of multiclass relevancy features, i.e., full class relevant (FCR) and partial class relevant (PCR) features. Particularly, FCR denotes genes that serve as candidate biomarkers for discriminating any cancer types. PCR, on the other hand, represents genes that distinguishes subsets of cancer types.

In this chapter, we propose a Markov blanket embedded multiobjective memetic algorithm (MBE-MOMA) [30] for simultaneous identification of both FCR and PCR genes in multiclass microarray cancer data. In particular, we consider an MBE-MOMA that searches on multiple objectives simultaneously with each objective corresponding to optimal PCR feature subsets. An approximate Pareto optimal set of solutions representing distinct options for solving the multiclass problem based on different tradeoffs of the objectives is obtained with a single simulation run of the MBE-MOMA. FCR genes are most likely to be selected among each approximate

Pareto optimal solution despite of the tradeoffs between the objectives. In contrast, PCR genes are selected in the extreme solution that optimizes a single corresponding objective.

Further, we study two strategies for the synergy of FCR and PCR feature subsets. The first ensemble strategy has been widely used in feature selection and classification [109, 110]. In this strategy, each of the FCR/PCR feature subset is used individually for classifying all classes, and the predictions of each trained classifier are then aggregated using a voting scheme. The second alternative is the conjunction strategy which is the union of all FCR and PCR feature subsets as a single feature subset. The union feature subset is then used for predicting the cancer types.

The remainder of this chapter is organized as follows: Section 6.2 presents our definitions of FCR and PCR features. Section 6.3 describes the details of the proposed MBE-MOMA algorithm. Section 6.4 presents the experimental results and discussions on both synthetic datasets and microarray datasets. Finally, Section 6.5 summarizes this study.

6.2 Multiclass Relevant Features

In this section, we present the definitions of relevant features, FCR and PCR features.

6.2.1 Multiclass Relevant Features

We begin with an overview on the categories of relevant features in multiclass problems. Let X be a full set of features, X_i be a feature, and $\overline{X}_i$ be the set of all features that excludes X_i , i.e., $\overline{X}_i = X - \{X_i\}$. An instance is typically described as an assignment of values $x = (x_1, \dots, x_N)$ to a set of features $X = (X_1, \dots, X_N)$ and a class label Y , which takes one of the k possible values in class set $C = \{c_1, \dots, c_k\}$. A value-assignment to all features in $\overline{X}_i$ is denoted by $\overline{x}_i$. Based on [11], features can be categorized into strongly relevant, weakly relevant, or irrelevant.

Definition 6.9 Strong Relevance: A feature X_i is strongly relevant if and only if there exists some x_i , c_j , and $\overline{x}_i$ for which $p(X_i = x_i, \overline{X}_i = \overline{x}_i) > 0$ such that $p(Y = c_j | X_i = x_i, \overline{X}_i = \overline{x}_i) \neq p(Y = c_j | \overline{X}_i = \overline{x}_i)$.

Definition 6.10 Weak Relevance: A feature X_i is weakly relevant if and only if it is not strongly relevant and there exists a subset of features $\overline{X}'_i \subseteq \overline{X}_i$ for which there exists some x_i , c_j , and $\overline{x}'_i$ for which $p(X_i = x_i, \overline{X}'_i = \overline{x}'_i) > 0$ such that $p(Y = c_j | X_i = x_i, \overline{X}'_i = \overline{x}'_i) \neq p(Y = c_j | \overline{X}'_i = \overline{x}'_i)$.

From the definitions in 6.9 and 6.10, a feature **relevant** to the learning classes C can either be strongly relevant or weakly relevant, otherwise, it is **irrelevant** to C .

6.2.2 Full/Partial Class Relevant

In cancer multiclass classification problems, besides identifying genes of relevance to C , it is also important for biologists to know which specific cancer or subset

of cancer types a gene is relevant to. Hence, it becomes necessary to extend the original definition of relevant feature to differentiate between partial class relevant (PCR) or full class relevant (FCR) features in multiclass problems. To define our notions of PCR and FCR features, we begin by first introducing the concept of Subset-Versus-Subset (SVS) Set.

Definition 6.11 *Subset-Versus-Subset Set:* A Subset-Versus-Subset set of classes C , $\mathcal{SVS}(C)$, is defined as $\mathcal{SVS}(C) = \{A, B\}$ where $A, B \subseteq C$ and $A \cap B = \emptyset$.

Definition 6.12 *Full Class Relevance:* A feature X_i is said to be full class relevant if and only if it is relevant to all possible SVS sets.

Definition 6.13 *Partial Class Relevance:* A feature X_i is said to be partial class relevant if and only if it is relevant to only some of the SVS sets, i.e., there exists a SVS set C' , such that X_i is irrelevant to C' .

FCR features are relevant to all SVS sets, including all pairs of classes $\{\{c_i\}, \{c_j\}\}$ ($i, j \in (1, \dots, k)$), they are important for distinguishing between any two classes. On the other hand, PCR features are relevant to only some of the SVS sets, this suggests PCR features are only important for distinguishing subset of classes but not all classes (A simple example illustrating FCR and PCR features is provided in Appendix C). An ideal solution for feature selection on multiclass problem is a FCR feature subset that can distinguish between any pair subsets of classes. However, in practice, it is always difficult to find such an ideal FCR feature subset that performs well on all classes. A selected feature subset performing well on some SVS sets

would not be successful on other SVS sets. Hence PCR features are indispensable for learning the multiclass problem.

We note that it can be computational expensive to find the true FCR and PCR features due to the large number of possible SVS sets. Hence, one key issue is to choose a scheme for generating SVS sets that can systematically cover the classes C . Since One-Versus-All (OVA) scheme has been shown to have superior performance than other strategy in the literature of multiclass cancer classification [76], we demonstrate the use of OVA scheme for generating the SVS sets in the present study. For classes $C = \{c_1, \dots, c_k\}$, OVA scheme creates k pairwise two-class SVS sets (labeled as **OVA sets** in this study), with each of them constructed as $\{c_i, \bar{c}_i\}$, where $i \in (1, \dots, k)$ and $\bar{c}_i = C - c_i$.

There exists an optimal PCR feature subset for each OVA set. A feature selected among all optimal PCR feature subsets is considered as a FCR feature. The straightforward way to identify all PCR and FCR features is to identify the optimal PCR feature subset for all OVA sets one by one, which takes k runs of feature selection algorithms. In this chapter, we propose a new multiobjective evolutionary algorithm, i.e., MBE-MOMA, for simultaneous identification of both FCR and PCR features in a single run. In particular, MBE-MOMA optimizes multiple objectives simultaneously with each objective corresponding to an OVA feature selection problem.

6.3 System and Methodology

Multiobjective evolutionary algorithms (MOEAs) have been widely used for feature selection [46, 47, 57, 58, 109], which solve a multiobjective problem (MOP) of minimizing the number of selected features and maximizing classification accuracy. In this chapter, we propose a novel MOEA for the simultaneous identification of both PCR and FCR features in multiclass problems, particularly, the Markov blanket embedded multiobjective memetic algorithm (MBE-MOMA).

6.3.1 Problem Formulation

Searching optimal PCR feature subsets for k OVA sets can naturally be casted as a MOP with each objective corresponds to the feature selection accuracy on each OVA set. Here, we solve this MOP using a Pareto based MOEA method. The MOP is thus defined as:

$$\begin{aligned} \min F(s) &= (f_1(s), \dots, f_k(s)) \text{ subject to } s \in S \\ f_i(s) &= -Acc(s, \{c_i, \bar{c}_i\}), (i \in (1, \dots, k)) \end{aligned} \quad (6.1)$$

where $F(s)$ is the objective vector, s is the candidate selected feature subset, k is the number of classes, and S is the feasible domain of s . The i -th objective $f_i(s)$ is then $Acc(s, \{c_i, \bar{c}_i\})$, which evaluates the classification accuracy on the i -th OVA set using the selected feature subset s .

Solving the MOP obtains an approximate Pareto optimal solutions set. The FCR

features can be identified as those common features found among all the approximate Pareto optimal feature subsets, since they are important for learning all OVA sets regardless of the tradeoff between the objectives. On the other hand, the optimal feature subset of an objective naturally forms the respective PCR features which are relevant to the corresponding OVA set. With a systematical abstraction from the approximate Pareto optimal set, it becomes possible to identify both FCR and PCR features simultaneously.

6.3.2 Markov Blanket Embedded Multiobjective Memetic Algorithm

In this subsection, we describe the proposed Markov blanket embedded multiobjective memetic algorithm (MBE-MOMA), which represents a synergy between the Pareto-based multiobjective evolutionary algorithm (MOEA) [56] and Markov blanket-based local search [29], for solving the MOP defined in Equation 6.1 as well as the simultaneous identification of FCR and PCR features. The pseudo code of the MBE-MOMA is outlined in Figure 6.1.

At the start of the MBE-MOMA search, an initial population of solutions is randomly generated with each chromosome encoding a candidate feature subset. In the present work, each chromosome is composed of a bit string of length equal to the total number of features in the feature selection problem of interest. Using binary encoding, a bit of '1' ('0') implies the corresponding feature is selected (excluded). The fitness of each chromosome is then obtained using an objective vector defined

**Markov Blanket Embedded Multiobjective Memetic Algorithm
BEGIN**

1. $t = 0$.
2. **Initialize:** Randomly generate an initial population $P(t)$ of feature subsets encoded with binary strings.
3. Evaluate fitness $F(s)$ of each solution in $P(t)$.
4. Rank $P(t)$ using Pareto dominance.
5. **While**(*Termination Criterion Not Fulfilled*)
 6. Select a temporary population $P'(t)$ from $P(t)$ based on Pareto rank.
 7. Perform crossover and mutation on $P'(t)$.
 8. Evaluate fitness $F(s)$ of each solution in $P'(t)$.
 9. Rank $\{P'(t) \cup P(t)\}$ using Pareto dominance.
 10. Apply Markov blanket based local search on the non-dominated solutions of $\{P'(t) \cup P(t)\}$ and generate an improved population $P''(t)$.
 11. Rank the population $\{P(t) \cup P'(t) \cup P''(t)\}$ using Pareto dominance.
 12. Distribute the population $\{P(t) \cup P'(t) \cup P''(t)\}$ based on a crowding distance [111].
 13. Select solutions from $\{P(t) \cup P'(t) \cup P''(t)\}$ to create a new population $P(t+1)$ based on Pareto ranking and crowding distance.
 14. $t = t+1$.
15. **End While**

END

Figure 6.1: Outline of Markov blanket embedded Multiobjective Memetic Algorithm for Feature Selection

in Equation 6.1.

In each MBE-MOMA generation, an offspring population $P'(t)$ is created from

the current population $P(t)$ using genetic operators, i.e., selection, crossover, and mutation. $P(t)$ and $P'(t)$ are merged together as a mating pool $\{P'(t) \cup P(t)\}$. Then a non-dominated sorting [56] is applied to classify the solutions in the mating pool into various Pareto fronts. The non-dominated solutions of $\{P'(t) \cup P(t)\}$ are selected to undergo the Markov blanket based LS, which has been described in Chapter 4.

MOEAs have shown promising performance in many complicate search problems. However, as discussed in Section 2.5, MOEAs may not be efficient when searching on problems with feature subset space having tens and thousands of genes. The hybridization of MOEAs and LS has been recently demonstrated to improve the searching ability of standard MOEAs. Such hybrid algorithms are more commonly known as MOMA. In this study, the MBE-MOMA is a hybridization of the standard MOEA with a Markov blanket based LS, since the latter is capable of dealing with irrelevant and redundant features efficiently.

The Markov blanket based LS used in MBE-MOMA is outlined in Figure 6.2. For a given candidate solution s , MBE-MOMA apply the LS to improve a randomly selected objective $f_i(s)$ based on the Add and Del operators and the improvement first strategy. In the improvement first strategy, a maximum computational complexity of l^2 is defined by the LS range l . For details of Add and Del operators, the reader is referred to Figure 4.2 and 4.3, respectively. Improvement is achieved when the new solution s' gives a higher accuracy on the selected objective than the original solution s , i.e., $f_i(s') > f_i(s)$. The new improved solution s' is archived in

Markov Blanket Based LS Using Improvement First Strategy
BEGIN

1. **For** Each non-dominated solution s .
2. Randomly select an objective $f_i(s)(i \in (1, \dots, k))$ to improve.
3. **For** $j = 1$ **to** l^2
4. Generate a unique random pair $\{a, d\}$ where $0 \leq a, d < l$.
5. Apply a times *Add* and d times *Del* on s to generate a new chromosome s' .
6. Calculate fitness of modified chromosome $F(s')$.
7. **IF** $f_i(s') > f_i(s)$
8. Break LS and return s' .
9. **End IF**
10. **End For**
11. **End For**

END

Figure 6.2: Markov blanket based LS in MBE-MOMA

a temporary population $P''(t)$. Afterward, a new population $P(t+1)$ of the same size as $P(t)$ is selected from $\{P(t) \cup P'(t) \cup P''(t)\}$. Elitism is ensured using Pareto ranking and the crowd distance [111] in $P(t+1)$.

The evolutionary operations used in MBE-MOMA include binary tournament selection [111], uniform crossover, and mutation operators [12]. Note that when two chromosomes are found having similar fitness on each objective (i.e., for a misclassification error of less than one data instance, the difference between their fitness on each objective is designed here to be less than a small value of $\varepsilon = 1/D$, where

D is the number of instances) the one with the smaller number of selected features is given a higher chance of surviving to the next generation. On the other hand, if prior knowledge on the optimum number of features is available, we constrained the number of bits '1' in each chromosome to a maximum of m in the evolutionary search process. In this case, the restrictive crossover and mutation [22] instead of the basic evolutionary operators are used, such that the number of bits '1' in each chromosome does not violate the constraint posed by the prior knowledge on m throughout the search.

6.3.3 Synergy Between FCR and PCR Feature Subsets

Since the output result of a Pareto-based MOEA is a set of approximate Pareto optimal solutions instead of only one optimal solution obtained in single objective EA, one key issue is how to select these solutions for practical use.

The common features appearing in all final solutions form the FCR feature subset. Those selected features (excluding the FCR features) in the optimum solution of each individual objective then form the PCR feature subset that is relevant to the corresponding OVA set. Hence, there are a total of $k + 1$ feature subsets including k subsets of PCR features, each relevant to a single OVA set, and 1 subset of FCR features relevant to all OVA sets.

In this study, we consider two strategies for combining these $k + 1$ feature subsets. The first ensemble strategy has been widely used in feature selection and classification [109,110]. In this strategy, each of the $k + 1$ feature subsets is used individually

for classifying all classes, and the prediction of each trained classifier is then aggregated using a voting scheme. The second alternative is the conjunction strategy which is the union of all $k + 1$ feature subsets in a single feature subset. The union feature subset is then used in estimating the classification accuracy.

The number of selected features in MBE-MOMA is then determined by the total number of unique features in all the $k + 1$ feature subsets. Since the $k + 1$ feature subsets are used rather than a single optimal feature subset in the single objective method, MBE-MOMA is expected to select more features than its single objective counterparts. Furthermore, relevant features that are not redundant in each of the $k + 1$ subsets could become redundant when used together. Hence, MBE-MOMA is also expected to select more redundant features than its single objective counterparts. Nevertheless, note that most of these redundant features selected in MBE-MOMA are actually PCR features that are important for distinguishing particular OVA sets. Further, it is also worth noting that the inclusion of these PCR features would help improve the classification accuracy and these PCR features could not be identified by using a single objective method counterpart.

6.4 Empirical Study

In this section, we study the performance of the proposed method MBE-MOMA using both synthetic and real world microarray datasets. The Markov blanket-embedded genetic algorithm (MBEGA) proposed in [29] and the standard multiobjective evolutionary algorithm (MOEA) based on NSGAII [111] are also used here for

comparison. MBEGA is the single objective counterpart of our MBE-MOMA where both share the same local search (LS) method, while MOEA is the multiobjective counterpart of MBE-MOMA without the LS.

The parameters for MBEGA, MOEA, and MBE-MOMA are configured similarly with population size of 50, crossover probability of 0.6, and mutation rate of 0.1. The LS range l in both MBEGA and MBE-MOMA is configured as in [29]. For fair comparison, the maximum computational budgets allowable for all these three algorithms are set to 2000 fitness functional calls. The configurations remain the same in our study on both the synthetic and microarray data. Based on prior knowledge, the maximum number of bit ‘1’ allowable in each chromosome is constrained to 50 and 150 for synthetic and microarray datasets, respectively.

To evaluate the performances of the feature selection algorithms considered in this study, the average of 30 independent runs of external .632 bootstrap [69] were reported. For some datasets, where some classes have very few instances, a stratified bootstrap sampling is applied instead. In particular, the sample distribution on each class is maintained consistent between the sampled and original datasets.

6.4.1 Synthetic Data

We begin with the use of synthetic multiclass datasets for studying the weaknesses and strengths of the feature selection algorithms as well as to illustrate the FCR and PCR features. Three three-class synthetic datasets are generated using the approach described in [112].

Synthetic dataset one (SD1) is designed to contain only FCR and irrelevant features. In particular, it is composed of 4020 genes and 75 samples with 25 samples for each class label. The centroids of the three classes are located at (0,0), (3.7,1), and (1,3.7). Two groups of relevant genes are generated from a multivariate normal distribution, with 10 genes in each group. The variance of each relevant gene is 1. All these 20 relevant genes are generated with the following mean and covariance matrixes:

$$\mu = \begin{bmatrix} \mu_0 & \mu_0 \\ \mu_{3.7} & \mu_1 \\ \mu_1 & \mu_{3.7} \end{bmatrix} \quad \sigma = \begin{bmatrix} \mathbf{e} & \mathbf{0} \\ \mathbf{0} & \mathbf{e} \end{bmatrix}$$

where $\mu_{\mathbf{x}}$ is a 25×10 matrix with each element equal to x ; $\mathbf{0}$ is a 10×10 matrix with each element equal to 0; $\mathbf{e}$ is a 10×10 symmetrical matrix with each diagonal element equal to 1 and the other elements equal to 0.9, i.e.,

$$\mathbf{e} = \begin{bmatrix} 1 & 0.9 & \dots & 0.9 \\ 0.9 & 1 & \dots & 0.9 \\ \vdots & \vdots & \ddots & \vdots \\ 0.9 & 0.9 & \dots & 1 \end{bmatrix}$$

The correlation between intra-group genes is 0.9, whereas the correlation between inter-group genes is 0. Genes in these two groups are FCR features, since they are relevant to any SVS sets of classes; and furthermore, genes in the same group are redundant with each other and the optimal gene subset for distinguishing the three classes consists of any 2 relevant genes from different groups. Another 4000

irrelevant genes are added to the data. Among these 4000 genes, 2000 are drawn from a normal distribution of $N(0,1)$ and the other 2000 genes are sampled with a uniform distribution of $U[-1,1]$.

Synthetic dataset two (SD2) is designed to contain FCR, PCR, and irrelevant features. It is composed of 4040 genes and 75 samples with 25 samples for each class label. The centroids of the three classes are located at $(0,1.18)$, $(3.7,1.18)$, and $(1,1.18)$ ¹. Four groups of relevant genes (G_0 , G_1 , G_2 , and G_3) are generated from a multivariate normal distribution, with 10 genes in each group. These 40 relevant genes are generated with the following mean and covariance matrixes:

$$\mu = \begin{bmatrix} \mu_0 & \mu_{1.18} & \mu_{-1.18} & \mu_{-1.18} \\ \mu_{3.7} & \mu_{-1.18} & \mu_{1.18} & \mu_{-1.18} \\ \mu_1 & \mu_{-1.18} & \mu_{-1.18} & \mu_{1.18} \end{bmatrix} \quad \sigma = \begin{bmatrix} e & 0 & 0 & 0 \\ 0 & e & 0 & 0 \\ 0 & 0 & e & 0 \\ 0 & 0 & 0 & e \end{bmatrix}$$

Genes in the same group are redundant to each other. Similar to SD1, another 4000 irrelevant genes have been included. In this dataset only genes in G_0 are FCR genes. While those genes in G_1 , G_2 , and G_3 are PCR genes. Further, genes in G_1 are drawn under different distributions for class 1 as compared to the other two classes, i.e., $N(1.18,1)$ for class 1 and $N(-1.18,1)$ for the remaining two classes, hence it is possible to distinguish class 1 from any other classes. However, since genes in G_1 are drawn under the same distribution of $N(-1.18,1)$ for samples in classes

¹The two coordinates(features) of the centroids for each class may not be the same. For example, the centroid of class 1 is specified on G_0 and G_1 , while the centroid of class 2 is specified on G_0 and G_2 .

2 and 3, it is difficult to distinguish between these two classes. The optimal gene subset to distinguish one class from others consists of one FCR gene from G0 and another PCR gene from the corresponding group. For instance, to correctly classify class 1, two genes each from G0 and G1 are required. The optimal gene subset to distinguish all the three classes thus consists of 4 genes, one FCR gene from G0 and three PCR genes each from G1, G2, and G3.

Last but not least, synthetic dataset three (SD3) has been designed to contain only PCR and irrelevant features. It is composed of 4060 genes and 75 samples with 25 samples for each class label. The centroids of the three classes are located at (-1.18,1.18), (1.18,-1.18), and (1.18,-1.18). Six groups of relevant genes (G0, G1, ..., G5) are generated from a multivariate normal distribution, with 10 genes in each group. These 60 relevant genes are generated with the following mean and covariance matrixes:

$$\mu = \begin{bmatrix} \mu_{-1.18} & \mu_{1.18} & \mu_{-1.18} & \mu_{1.18} & \mu_{-1.18} & \mu_{1.18} \\ \mu_{1.18} & \mu_{-1.18} & \mu_{1.18} & \mu_{-1.18} & \mu_{-1.18} & \mu_{1.18} \\ \mu_{1.18} & \mu_{-1.18} & \mu_{-1.18} & \mu_{1.18} & \mu_{1.18} & \mu_{-1.18} \end{bmatrix} \quad \sigma = \begin{bmatrix} e & 0 & 0 & 0 & 0 & 0 \\ 0 & e & 0 & 0 & 0 & 0 \\ 0 & 0 & e & 0 & 0 & 0 \\ 0 & 0 & 0 & e & 0 & 0 \\ 0 & 0 & 0 & 0 & e & 0 \\ 0 & 0 & 0 & 0 & 0 & e \end{bmatrix}$$

Genes in the same group are designed to be redundant to each other. Like SD1 and SD2, additional 4000 irrelevant genes are also included. All the relevant features

are PCR features and the optimal gene subset for distinguishing one class from the others consists of two PCR genes each from the corresponding group. For instance, two genes each from G0 and G1 form the optimal feature subset for separating class 1 and the other two classes. The optimal gene subset to distinguish all the three classes thus consists of 6 genes with one from each group.

Feature Selection Results on Synthetic Datasets

The MBEGA, MOEA, and MBE-MOMA algorithms are used to search on each of the three synthetic datasets (i.e., SD1, SD2, and SD3) and the classification accuracies obtained are reported in Table 6.1. It is observed that on the datasets with PCR features, i.e., SD2 and SD3, MBE-MOMA using both the ensemble and conjunction strategies outperforms all the other algorithms considered. Overall, the ensemble MBE-MOMA generates the best classification accuracies on all three datasets.

Table 6.1: Classification accuracy by each algorithm on synthetic data

	MBEGA	MOEA-E*	MOEA-C*	MBE-MOMA-E*	MBE-MOMA-C*
SD1	86.88	84.11	78.66	90.83	85.97
SD2	90.94	85.23	83.04	94.55	92.09
SD3	95.49	88.37	89.39	98.26	97.35

* -E and -C denote the algorithms using ensemble and conjunction strategy, respectively.

The list of corresponding selected features by all three algorithms is summarized in Table 6.2. The results suggest that MBE-MOMA selects more features among the optimal subset than the other algorithms. On SD1, all three algorithms have successfully identified the 2 FCR features among the optimal subset. On the other

two datasets, only MBE-MOMA manages to identify the 4 FCR and PCR features among the optimal set hence producing superior classification accuracy over MBEGA and MOEA. In comparison, MBEGA selects least features among the optimal subset, but it also eliminates more irrelevant and redundant features than the other two algorithms. Besides requiring a larger number of features to arrive at the poorer classification accuracy on the datasets, MOEA also fails to remove the irrelevant features due to the lack of the Markov blanket-based LS.

In summary, the results on synthetic datasets also indicate that the classification accuracy deteriorates when some features in the optimal subset are missed out and/or irrelevant features are selected. In addition, both multiobjective approaches, MOEA and MBE-MOMA, are capable of identifying PCR features that the single objective approach, MBEGA, fails to locate.

Table 6.2: Feature selected by each algorithm on synthetic data

	Features	MBEGA	MOEA	MBE-MOMA
SD1	(#)	8.2	57.5	25.4
	$OPT(2)$	2	2	2
	<i>Red</i>	3.3	3.2	7.0
	<i>Irr</i>	2.9	52.3	16.4
SD2	(#)	6.6	61.3	19.3
	$OPT(4)$	3.3	3.8	4
	<i>Red</i>	2.4	3.1	7.2
	<i>Irr</i>	0.9	54.4	8.1
SD3	(#)	11.9	58.1	21.6
	$OPT(6)$	5.3	5.6	5.9
	<i>Red</i>	5.2	3.6	10.1
	<i>Irr</i>	1.4	48.9	4.6

(#): Number of Selected features; $OPT(x)$: Number of selected features with in the optimal subset, x indicates the optimal number of features; *Red*: Redundant Features; *Irr*: Irrelevant Features.

The FCR and PCR Features in Final Solution

In this section, we illustrate the FCR and PCR features in the final solutions. Taking SD2 as an example, since it contains both FCR and PCR features, the Pareto front obtained by MBE-MOMA is plotted in Figure 6.3. Each point on this front presents the corresponding selected feature subset in the approximate Pareto optimal set. Since FCR features are crucial for learning every OVA set, they should be selected in all solutions. As expected, we can find at least one FCR feature, i.e., a feature belonging to group G0, in each approximate Pareto optimal solution.

Figure 6.3: The three-dimension Pareto front on the SD2

To further illustrate the PCR features, we project the 3-dimensional Pareto front on each of the two objectives (shown in Figure 6.4). For example, the fitness values of objective 1 (class 1 vs. others) and objective 2 (class 2 vs. others) for each approximate Pareto optimal solution are depicted in Figure 6.4(a). In SD2, G1 and G2 are two groups of PCR genes relevant to objective 1 and 2, respectively.

In Figure 6.4(a), solutions that select 3 times more genes from G2 than G1 are represented as “ Δ ”, while solutions selecting 2 times more genes from G1 than G2 are denoted as “ $\square$ ”. Solutions containing similar number of genes of G1 and G2 are marked with “o”. Note that solutions marked with “ Δ ” and “ $\square$ ” contain very few irrelevant features, suggesting most of features (except for those FCR features that exist in every solution) selected in these two kinds of solutions are PCR features relevant to the corresponding objective. For example, most features selected in “ Δ ” solutions are PCR features from G2 that are relevant to objective 2. Furthermore, almost all the “ Δ ” solutions plotted in Figure 6.4(a) select exactly the same PCR features from G2. Therefore, one among these “ Δ ” solutions would be sufficient for identifying PCR feature relevant to objective 2. Similarly, Figure 6.4(b) and Figure 6.4(c) illustrate the solutions with respect to objective 1 vs. objective 3 and objective 2 vs. objective 3, respectively. In this study, we select the extremal solution to represent the PCR features of each objective, since it gives the optimal fitness value of each objective.

6.4.2 Microarray Data

We further investigate the performance of the proposed algorithm using 10 real world multiclass microarray datasets described in Table 6.3.

Classification Accuracy

Table 6.4 reports the average classification accuracy obtained by each feature selection algorithm on the ten datasets over 30 runs of .632 bootstraps. For each

Figure 6.4: Two-dimension projections of the Pareto front. (a) Objective 1 vs. Objective 2, (b) Objective 1 vs. Objective 3, (c) Objective 2 vs. Objective 3. $Gx \ll Gy$ ($x, y = 1, 2, 3$) indicates more genes are selected from group y than x . $Gx \approx Gy$ ($x, y = 1, 2, 3$) indicates similar number of genes are selected from group x and y ,

dataset, we also report the percentage of FCR features against the total selected features made by MBE-MOMA.

Table 6.3: Description of 10 Multiclass Microarray Datasets

Dataset	Genes	Samples	Classes	Description and Reference
ALL-AML-3	7129	72	3	AML, ALL B-cell, and ALL T-Cell. [89]*
ALL-AML-4	7129	72	4	AML-BM, AML-PB, ALL B-cell, and T-Cell. [89]**
Lymphoma	4026	62	3	Three most prevalent adult lymphoid tumors. [113]
MLL	12582	72	3	AML, ALL, and mixed-lineage leukemia (MLL). [114]
SRBCT	2308	83	4	Small, round blue cell tumors of childhood. [86]
NCI60	6114	61	9	61 cell lines from 9 human cancer types. [16, 17, 115]
Brain	10367	50	4	4 malignant glioma types. [116]
Thyroid	2000	168	4	3 thyroid tumor types (FA, FC, and PC) and 1 normal tissues (N). [117]***
ALL	12558	248	6	6 subtypes of ALL. [118]
Lung5c	12600	203	5	4 lung cancer types and 1 normal tissues. [119]

*AML: Acute Myelogenous Leukemia; ALL: Acute Lymphoblastic Leukemia.

**BM: Bone Marrow samples; PB: Peripheral Blood samples.

***FA: Follicular adenoma; FC: Follicular carcinoma; PC: Papillary Carcinoma.

Table 6.4: Classification accuracy of feature selection algorithms on ten multiclass microarray datasets

	FCR%	MBEGA	MOEA-E	MOEA-C	MBE-MOMA-E	MBE-MOMA-C
ALL-AML-3	30.4	96.64	94.38	94.67	96.61	96.49
ALL-AML-4	13.1	91.93	91.53	90.89	89.53	93.02
Lymphoma	7.7	97.68	98.74	98.72	98.60	98.29
MLL	25.0	94.33	95.14	93.82	96.11	95.45
SRBCT	0.4	99.23	98.46	98.52	98.14	99.43
NCI60	0	65.58	73.56	69.94	69.43	69.06
Brain	1.5	72.53	79.93	77.65	82.65	77.75
Thyroid	0.7	78.14	81.86	79.63	81.87	80.67
ALL	0	95.25	94.76	94.95	97.24	97.40
Lung5c	0.8	88.54	95.23	95.55	94.97	95.83
$\overline{Acc}^*$		87.99	90.36	89.43	90.51	90.34
$\overline{\#gene}^{**}$		32.5	269.8	269.8	118.5	118.5

* $\overline{Acc}$: Average of classification accuracy over all datasets.

** $\overline{\#gene}$: Average number of selected genes over all datasets.

In Table 6.4, note that the classification accuracies obtained by all the algorithms are not significantly different on the first 5 microarray datasets, because these datasets have higher percentages of FCR (except for the SRBCT), making them easy for most feature selection classification algorithms in discriminating the classes. On

the other 5 datasets, both MOEA and MBE-MOMA outperform MBEGA significantly, due to the low or zero percentage of FCR features available. Since these 5 datasets contain mainly PCR features, it is difficult for single-objective methods, including MBEGA, to identify them. Further, MBE-MOMA selects much less genes at competitive or better classification accuracies than MOEA, due to the use of Markov blanket based LS. Overall, MBE-MOMA with ensemble strategy yields the best average accuracy over all the datasets.

Selected FCR and PCR Genes

One core strength of MBE-MOMA is in its ability to identify both FCR and PCR genes simultaneously. In this subsection, we validate the FCR and PCR genes selected by MBE-MOMA. In particular, we focus on the ALL-AML-3 and Thyroid datasets as representatives datasets having high and low percentages of FCR genes, respectively.

ALL-AML-3 represents one of datasets that have been extensively studied by many researchers. Hence, there is more information in the literature that may allow one to truly validate the selected genes. Table 6.5 tabulates the 20 most selected FCR genes on ALL-AML-3 dataset. Note that 6 out of these 20 genes, i.e., U05259, X95735, M23197, M31523, M83652, and M84526, were identified as important biomarkers for distinguishing the classes AML and ALL, in [89]. X95739 and M84526 were also identified as distinguishing genes for AML and ALL by Zhou et al. [85]. Nevertheless, as MBE-MOMA considers a 3-class (AML, ALL T-cell, and ALL B-cell) multiobjective problem, as opposed to a two-class problem in [85, 89],

the result obtained in the present study appears to suggest all six identified genes in [89] to express differently in the subtypes of ALL, i.e., ALL T-cell and ALL B-cell. Besides these 6 genes, two other genes, X03934 and M27891, identified by MBE-MOMA are consistent to that also selected by Yeung et al. [108].

Table 6.5: 20 most selected FCR genes on the ALL-AML-3 dataset

M84371_rna1_s	CD19 gene
U05259_rna1	MB-1 gene
D00749_s	T-cell antigen CD7 precursor
X00437_s	TCRB T-cell receptor, beta cluster
X95735	Zyxin
M23323_s	T-cell surface glycoprotein CD3 epsilon chain precursor
M37271_s	T-cell antigen CD7 precursor
U46499	Glutathione S-transferase, microsomal
U23852_s	GB DEF = T-lymphocyte specific protein tyrosine kinase p56lck (lck) abberant mRNA
M89957	IGB Immunoglobulin-associated beta (B29)
M23197	CD33 CD33 antigen (differentiation antigen)
M31523	TCF3 Transcription factor 3 (E2A immunoglobulin enhancer binding factors E12/E47)
D11327_s	PTPN7 Protein tyrosine phosphatase, non-receptor type 7
X03934	GB DEF = T-cell antigen receptor gene T3-delta
M83652_s	PFC Properdin P factor, complement
M84526	DF D component of complement (adipsin)
X58529	IGHM Immunoglobulin mu
L09209_s	APLP2 Amyloid beta (A4) precursor-like protein 2
X76223_s	GB DEF = MAL gene exon 4
M27891	CST3 Cystatin C (amyloid angiopathy and cerebral hemorrhage)

Besides FCR genes, we also report the 10 most selected PCR genes for each of the 3 OVA sets on ALL-AML-3 dataset in Tables 6.6 to 6.8. Among the selected PCR genes, X77094 is found to be relevant to the first (ALL B-cell vs. others) and second (ALL T-Cell vs. others) OVA sets, suggesting it as a crucial biomarker for distinguishing ALL B-cell and ALL T-cell samples. Genes D88270 and L08895 are

consistently found as relevant to the first (ALL B-cell vs. others) and third (AML vs. others) OVA sets, suggesting they represent important genes for studying AML and ALL B-cell cancer types. Indeed, they have been demonstrated to be potential biomarkers for discriminating subtypes of leukemia in [120,121]. Last but not least, genes M31303 and M31211 listed in Table 6.8 are also identified as biomarkers for distinguishing AML and ALL by Golub et al. [89].

Table 6.6: 10 most selected PCR genes relevant to (ALL B-cell vs. Others) on the ALL-AML-3 dataset

D88270	GB DEF = (lambda) DNA for immunoglobulin light chain
L08895	MEF2C MADS box transcription enhancer factor 2, polypeptide C (myocyte enhancer factor 2C)
U31248	ZNF174 Zinc finger protein 174
M96326_rna1	Azurocidin gene
U09578	MAPKAP kinase (3pK) mRNA
X99920	S100 calcium-binding protein A13
X82240_rna1	TCL1 gene (T cell leukemia) extracted from H.sapiens mRNA for Tcell leukemia/lymphoma 1
M22960	PPGB Protective protein for beta-galactosidase (galactosialidosis)
X77094	P40phox
U51240	KIAA0085 gene, partial cds

Next we consider the Thyroid dataset, which is a relatively new dataset first published in 2006 [117]. It is hoped that this study on the Thyroid dataset will provide some useful prediction to assist biologists in the discovery of new biomarkers for thyroid cancer. Only 5 FCR (tabulated in Table 6.9) are found by MBE-MOMA in this dataset. Ten most selected PCR genes for each of the 4 OVA sets are listed in Tables 6.10 to 6.13. Among these selected PCR genes, LT55 is found to be crucial for learning Follicular adenoma and Normal tissues, since it is always

Table 6.7: 10 most selected PCR genes relevant to (ALL T-Cell vs. Others) on the ALL-AML-3 dataset

HG4128- HT4398	Anion Exchanger 3, Cardiac Isoform
X59871	TCF7 Transcription factor 7 (T-cell specific)
X00274	HLA class II histocompatibility antigen, DR alpha chain precursor
U14603	Protein tyrosine phosphatase PTPCAAX2 (hPTPCAAX2) mRNA
X62744	Class II histocompatibility antigen, M alpha chain precursor
U18548	GB DEF = GPR12 G protein coupled-receptor gene
U46006_s	GB DEF = Smooth muscle LIM protein (h-SmLIM) mRNA
Z35227	TTF mRNA for small G protein
X82240_rna1	TCL1 gene (T cell leukemia) extracted from H.sapiens mRNA for Tcell leukemia/lymphoma 1
X77094	P40phox

Table 6.8: 10 most selected PCR genes relevant to (AML vs. Others) on the ALL-AML-3 dataset

D88270	GB DEF = (lambda) DNA for immunoglobulin light chain
L08895	MEF2C MADS box transcription enhancer factor 2, polypeptide C (myocyte enhancer factor 2C)
M96326_rna1	Azurocidin gene
M63138	CTSD Cathepsin D (lysosomal aspartyl protease)
M31303_rna1	Oncoprotein 18 (Op18) gene
M92287	CCND3 Cyclin D3
M62762	ATP6C Vacuolar H+ ATPase proton channel subunit
M31211_s	MYL1 Myosin light chain (alkali)
D88422	CYSTATIN A
J05243	SPTAN1 Spectrin, alpha, non-erythrocytic 1 (alpha-fodrin)

selected for distinguishing the OVA sets of (Follicular adenoma vs. Others) and (Normal vs. Others). For the same reason, gene GS2252 is important for studying Follicular carcinoma and Papillary Carcinoma tissues, since it is always selected for distinguishing the OVA sets of (Follicular carcinoma vs. Others) and (Papillary Carcinoma vs. Others).

Table 6.9: 5 selected FCR genes on the Thyroid dataset

LT36	keratin 19
GS421	Homo sapiens thymosin, beta 4, X chromosome (TMSB4X), mRNA.
GS8241	Homo sapiens protein phosphatase 1, regulatory (inhibitor) subunit15A (PPP1R15A), mRNA.
GS3295	Homo sapiens trefoil factor 3 (intestinal) (TFF3), mRNA.
GS3883	Homo sapiens cytochrome c oxidase subunit VIII (COX8), nuclear geneencoding mitochondrial protein, mRNA.

Table 6.10: 10 most selected PCR genes relevant to (Follicular adenoma vs. Others) on the Thyroid dataset

LT55	G0S3 mRNA
GS15425	Homo sapiens splicing factor, arginine/serine-rich 2 (SFRS2), mRNA.
GS2257	Homo sapiens thymosin, beta 10 (TMSB10), mRNA.
GS14741	Homo sapiens thyroid peroxidase (TPO), nuclear gene encoding mitochondrial protein, mRNA.
LT62	apolipoprotein D mRNA
GS4484	Homo sapiens KIAA0089 protein (KIAA0089), mRNA.
GS1213	Homo sapiens apoptosis inhibitor 5 (API5), mRNA.
GS13671	Homo sapiens metallothionein 1L (MT1L), mRNA.
LT38	adenosine A1 receptor (ADORA1) mRNA exons 1-6
GS2252	Homo sapiens serine (or cysteine) proteinase inhibitor, clade A(alpha-1 antiproteinase, antitrypsin), member 1(SERPINA1), mRNA.

Comparison to Other Feature Selection Studies in the Literature

Comparing the results from this study with others published in the literature may not be suitable due to the different data preprocessing, classifier, performance evaluation schema, experiment design etc. used in different studies. However, the reported results in the literature confirm that MBE-MOMA performs competitively or better than the previously published methods for the same datasets.

Table 6.11: 10 most selected PCR genes relevant to (Follicular carcinoma vs. Others) on the Thyroid dataset

GS2266	Homo sapiens transmembrane 4 superfamily member 1 (TM4SF1), mRNA.
GS2646	-
LT96	CBFB(core-binding factor, beta subunit, isoform 1)
GS6511	-
GS3909	-
GS6487	Homo sapiens fatty-acid-Coenzyme A ligase, long-chain 4 (FACL4),transcript variant 1, mRNA.
GS110	Homo sapiens ribosomal protein L21 (RPL21), mRNA.
GS7292	Homo sapiens activating transcription factor 3 (ATF3), mRNA.
GS14049	Homo sapiens paraneoplastic antigen MA1 (PNMA1), mRNA.
GS9988	Homo sapiens hypothetical protein FLJ21562 (FLJ21562), mRNA.

Table 6.12: 10 most selected PCR genes relevant to (Papillary Carcinoma vs. Others) on the Thyroid dataset

GS453	Homo sapiens cytochrome c oxidase subunit VIIc (COX7C), nucleargene encoding mitochondrial protein, mRNA.
LT57	thyroid peroxidase mRNA
GS4923	Homo sapiens tissue inhibitor of metalloproteinase 1 (erythroidpotentiating activity, collagenase inhibitor) (TIMP1),mRNA.
LT29	chitinase 3-like 1
GS2252	Homo sapiens serine (or cysteine) proteinase inhibitor, clade A(alpha-1 antiproteinase, antitrypsin), member 1(SERPINA1), mRNA.
GS3294	Homo sapiens short-chain dehydrogenase/reductase 1 (SDR1), mRNA.
LT156	stromelysin-3 precursor=MMP11
LT139	CTSH(cathepsin H isoform a preproprotein)
LT41	melanocyte-specific gene 1 (msg1)
GS6511	-

For example, Li et al. [87] reported their investigation results on the ALL-AML-3, ALL-AML-4, MLL, ALL, Lymphoma, SRBCT, and NCI60 datasets using various classifiers and feature selection algorithms and 4-fold external cross validation. The

Table 6.13: 10 most selected PCR genes relevant to (Normal vs. Others) on the Thyroid dataset

LT55	G0S3 mRNA
GS1418	Homo sapiens cathepsin B (CTSB), transcript variant 5, mRNA.
LT41	melanocyte-specific gene 1 (msg1)
GS453	Homo sapiens cytochrome c oxidase subunit VIIc (COX7C), nucle- argene encoding mitochondrial protein, mRNA.
GS4210	Homo sapiens Niemann-Pick disease, type C2 (NPC2), mRNA.
GS15881	Homo sapiens fibulin 2 (FBLN2), mRNA.
GS1329	Homo sapiens v-fos FBJ murine osteosarcoma viral oncogene ho- molog(FOS), mRNA.
GS2677	Homo sapiens NADH dehydrogenase (ubiquinone) 1 beta subcom- plex, 2,8kDa (NDUFB2), mRNA.
LT53	CL 100 mRNA for protein tyrosine phosphatase
LT62	apolipoprotein D mRNA

best classification accuracies obtained on the first 5 datasets are competitive with our results, however 150 genes were required to produce these competitive accuracies, which is significantly more than the average of 118.5 genes identified by MBE-MOMA. On the NCI60 dataset, MBE-MOMA obtains an average accuracy of 69.43%, which is also superior to the best accuracy of 66.66% reported in [87].

Ooi et al. [105] proposed a new feature selection method for multiclass gene selection based on the degree of differential prioritization and F-splits evaluation procedure. On datasets, NCI60, Lung5c, SRBCT, MLL, and ALL-AML-3, their new method obtained best accuracies of 68.0%, 93.8%, 98.9%, 97.9%, and 97.1%, respectively. In comparison, MBE-MOMA gives competitive accuracy on these five datasets, with higher accuracies on the first three datasets while lower accuracies on the other two. Note that both methods have selected similar number of genes.

Ooi et al. [16] and Liu et al. [17] proposed two GAs based methods, GA with

maximum likelihood and GA with SVM, for multiclass gene selection, respectively. Both methods were applied to NCI60 dataset with 1000 pre-selected genes and the results are reported in [17]. GA with SVM was reported to outperform GA with maximum likelihood with a leave-one-out cross validation accuracy of 88.52% using 40 genes. Unfortunately, the results were obtained using an internal cross validation, which suffers from selection bias [74] or overfitting [73]. In Chapter 4, we have compared MBEGA with GA/SVM using the external .632 bootstrap. MBEGA was shown to attain a better accuracy of 70.38% with 47.6 selected genes, while GA/SVM obtains an accuracy of only 65.79% with 40 selected genes. On the other hand, MBE-MOMA arrived at a significantly better accuracy of 74.47% using 120.3 genes on the same dataset.

On the Thyroid dataset, Yukinawa et al. [117] used two strategies, i.e., one class vs. any subset of other class and any pairs of disjoint subsets of classes, for constructing a multiclass predictor from binary classifier. They reported the best accuracy of 79.8% using leave-one-out cross validation, while MBE-MOMA arrived at a significantly better accuracy of 81.87%.

6.5 Summary

In this chapter, we have defined two new types of relevant features, i.e., full class relevant (FCR) and partial class relevant (PCR), for multiclass problems, and also proposed a novel Markov blanket embedded multiobjective memetic algorithm (MBE-MOMA) for simultaneous identification of FCR and PCR features. Empirical stud-

ies on three synthetic and ten microarray datasets suggests that MBE-MOMA is efficient in identifying both FCR and PCR features and yields good classification accuracy. It performs better than the single objective counterpart, MBEGA, as well as standard MOEA without using LS. MBE-MOMA also produces superior or competitive performance over other gene selection methods published in the literature. Through this study, it is the authors' hope that the proposed MBE-MOMA serves to enhance the process of candidate biomarkers discovery as well as assist researchers in analyzing the growing amounts of biological research data, especially on multiclass problems.

Chapter 7

Conclusions

Feature selection remains an active research area in the communities of pattern recognition, data mining and machine learning. Though there have been numerous efforts dedicated to it, still this problem is far from being completely solved. In this research, we have proposed feature selection algorithms using MA, which provide a novel efficient searching method on a huge feature space and also a novel hybrid schema for complete advantages of both filter and wrapper methods. In this chapter, we summarize the contributions of this research and also describe some possible future directions.

7.1 Contributions

In this research, we proposed new memetic frameworks for synergy between filter and GA wrapper methods. In particular, filter methods are introduced to fine-tune the population of GA solutions by adding or deleting features. These new hybrid algorithms provide efficient searching methods for feature selection problems where the feature size is extremely large, such as microarray data.

Three novel feature selection algorithms using MA have been presented and

systematically studied in this dissertation. In what follows, we summarize three novel MAs for feature selection.

1. **Wrapper-Filter Feature Selection Algorithm (WFFSA)** [22]: WFFSA fine-tunes the population of GA solutions by adding or deleting features based on an univariate feature filter ranking method (e.g., ReliefF, Gain Ratio, or Chi-square). Empirical studies on UCI and microarray datasets suggest WFFSA searches more efficiently and is capable of producing a good classification accuracy with a small number of features simultaneously. It outperforms GA, MA with sequential LS, as well as many existing algorithms in the literature.
2. **Markov Blanket Embedded Genetic Algorithm (MBEGA)** [29]: MBEGA embeds the Markov blanket to fine-tune the search by adding relevant features and removing redundant and/or irrelevant features in the GA solutions. In contrast to WFFSA, MBEGA is targeted to handle problems with much more irrelevant and redundant features, i.e., microarray data. Comparative studies on synthetic and microarray datasets suggest MBEGA is able to eliminate irrelevant and redundant features efficiently using the Markov blanket and the predictive power of the wrapper model. It attains better accuracy and robustness than other feature selection algorithms considered in a reasonable time. It also achieves the best compromise on all the four evaluation criteria, i.e., classification accuracy, number of selected genes, computational cost, and robustness.

3. Markov Blanket Embedded Multiobjective Memetic Algorithm (MBE-

MOMA) [30]: MBE-MOMA represents a synergy between MOEA and Markov blanket based LS. It is designed for the simultaneous identification of both full class relevant (FCR) and partial class relevant features (PCR) in multiclass problems. MBE-MOMA is demonstrated to be significantly more efficient in the identification of both FCR and PCR genes on multiclass microarray datasets than the single objective MBEGA counterpart and MOEA without the Markov blanket-based LS. In particular, in terms of classification accuracy, it scores better than or competitive to MBEGA and MOEA, as well as some existing methods in the literature, especially on datasets with a small percentage of FCR features.

This dissertation represents one of the earliest research efforts on MA for feature selection. Various LS methods, strategies for applying LS, and LS intensities for feature selection problems have been investigated in this research. The results and observations obtained in this dissertation provide valuable guidelines for further research in this area.

7.2 Future Work

We have made promising progress on studying MA for feature selection. Nevertheless, we believe there remains plenty of room for future study. Here we discuss some possible immediate extensions of this research.

Simultaneous analysis of diverse data types: Microarray has posed as an

excellent tool for predicting cancer type and gene function in a high-throughput fashion. However, microarray data often lacks the degree of specificity needed for accurate prediction due to some inherent limitations. For instance, genes that are regulated at the translational level or protein level cannot be expressed using microarray technologies. Furthermore, genes with corrupted microarray data may lead to misclassification since the data collection process is subjected to human and instrumental errors.. To address these problems, other data types (e.g. genome sequence, protein expression profile, protein-protein interaction, protein-DNA interactions, transcription factor binding sites data, etc) may be incorporated. Hence, it would be interesting to extend the gene selection algorithms for simultaneously analyzing of diverse data types.

Enhance search efficiency: The application of evolutionary computation techniques for optimization problems with high evaluation cost is an increasingly important area of research. Although it has been established that evolutionary computation techniques are powerful optimization tools, researchers are facing the challenge of increasing computational cost of today's applications. In this work, the computational cost of MA feature selection increases with the feature size and the number of classification calls involved in the optimization process, which may make the search prohibitive or impractical without the use of high performance computing resources or computationally cheap surrogate models. Hence another future direction is to reduce the computational cost consumed by the classification with more efficient and robust classifiers or surrogate models.

Automate the process of gene validation: It is worth noting that the validation of the selected genes remains to be a manual driven and subjective process. An immediate future work is to incorporate domain knowledge obtained from existing biology literature and online databases relating to gene selection. This will serve to validate the selected genes automatically and guide the search towards the optimal gene feature set.

Author's Publications

Journal Papers

1. Zexuan Zhu, Y. S. Ong and M. Dash, "Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework," *IEEE Transactions on Systems, Man and Cybernetics - Part B*, Vol. 37, No. 1, pp. 70-76, 2007
2. Zexuan Zhu, Y. S. Ong and M. Dash, "Markov Blanket-Embedded Genetic Algorithm for Gene Selection," *Pattern Recognition*, Vol. 40, No. 11, 3236-3248, 2007
3. Zexuan Zhu, Y. S. Ong and J. M. Zurada, "Simultaneous Identification of Full Class Relevant and Partial Class Relevant Genes," *BMC Bioinformatic*, 2007 (Submitted)

Conference Papers

1. S. K. Ng, Zexuan Zhu and Y. S. Ong, "Whole-Genome Functional Classification of Genes by Latent Semantic Analysis on Microarray Data," *2nd Asia-*

Pacific Bioinformatics Conference (APBC2004), Dunedin, New Zealand, 18 - 22 Jan 2004.

2. Zexuan Zhu, Y. S. Ong, K. W. Wong and K. T. Seow, "Experimental Condition Selection In Whole-Genome Functional Classification," *IEEE Conference on Cybernetic and Intelligent System (CIS2004)*, Singapore, December 2004
3. Zexuan Zhu and Y. S. Ong, "Memetic Algorithms For Feature Selection On Microarray Data", *4th International Symposium on Neural Networks (ISNN2007)*, Nanjing, China, 3-7 Jun 2007

Appendix A

Filter Feature Selection Methods

A.1 ReliefF Feature Selection

ReliefF [24] is an extension of Relief [24] and it is more robust because it can deal with multiclass problems as well as incomplete and noisy data. ReliefF randomly selects an instance I_i , and then searches for k of its nearest neighbors from the same class, called the nearest hits H_j , and also k nearest neighbors from each of the different classes, called nearest miss $M_j(C)$. It updates the quality estimation $W[F]$ for all features F based on an intuitive idea that a feature is more relevant if it distinguishes between an instance and its nearest misses $M_j(C)$, and less relevant if it distinguishes between an instance and its nearest hits H_j . The update formula is defined as follow:

$$W[F] = W[F] - \sum_{j=1}^k Diff(F, I_i, H_j)/(s \cdot k) + \sum_{C \neq class(I_i)} \left[\frac{P(C)}{1 - P(class(R_i))} \sum_{j=1}^k Diff(F, I_i, M_j(C)) \right] / (s \cdot k) \quad (A.1)$$

where $P(C)$ is the prior probability of class C . $Diff(F, I_1, I_2)$ calculates the difference between the values of the feature F for two instances I_1 and I_2 . After exhausting all instances in the sample, we rank all the features based on their weight. The algorithm of ReliefF is described in Figure A.1.

Relief
BEGIN

1. Set all weights $W[F] = 0$;
2. **For** ($i = 1$ to s) *//the procedure is repeated s times*
3. Randomly select an instance I_i ;
4. Find k nearest hits H_j ;
5. **For** each class $C \neq class(I_i)$
6. Find k nearest misses $M_j(C)$;
7. **End For**
8. **For** ($f = 1$ to $|F|$)
9. Update $W[f]$ according to Equation A.1;
10. **End For**
11. **End For**
12. Sort the features according to $W[F]$;

END

Figure A.1: Outlines of ReliefF

A.2 Gain Ratio Feature Selection

Gain ratio feature selection [25] evaluates features individually by measuring their gain ratio with respect to the classes. For continuous values, the method first dis-

cretizes the data using Fayyad's information theoretic binning [99]. The gain ratio of a feature is defined as:

$$GR(C, X) = \frac{I(C) + I(X) - I(C, X)}{I(X)} \quad (\text{A.2})$$

where:

$$I(C) = - \sum_{i=1}^{|C|} P(C_i) \log P(C_i) \quad (\text{A.3})$$

$$I(X) = - \sum_x P(X = x) \log P(X = x) \quad (\text{A.4})$$

$$I(C, X) = - \sum_x P(X = x) \sum_{i=1}^{|C|} P(C_i, X = x) \log P(C_i, X = x) \quad (\text{A.5})$$

$P(C_i)$ is the probability of an instance belonging to class C_i . $P(X = x)$ denotes the probability of finding instances with feature value $X = x$, or within the discretized interval $X \in x$. $P(C_i, X = x)$ represents the joint probability of finding the feature value $X = x$ for the instance belonging to class C_i . $I(C, X)$ calculates the information contained in the joint distribution of classes and features, summed over all classes. A feature is more important if its gain ratio is larger.

A.3 Chi Square Feature Selection

Another method we use to rank features is Chi-Square method [26], which evaluates features individually by measuring their chi-square statistic with respect to the classes. Continuous values are discretized using Fayyad's method [99]. The

chi-square value of a feature is defined as:

$$\chi^2(X) = \sum_{i=1}^s \sum_{j=1}^k \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \quad (\text{A.6})$$

$$E_{ij} = I_i \cdot C_j / N \quad (\text{A.7})$$

where s is the number of discretized intervals, k is the number of classes, A_{ij} denotes the number of instances that belong to j -th class and whose value of feature X falls in the i -th interval, I_i is the number of instances in the i -th interval, C_j is the number of instances in the j -th class, N represents the total number of instances, and E_{ij} is the expected frequency of A_{ij} . The larger the chi-square value, the more important the feature is.

Appendix B

Classification Methods

B.1 K-nearest Neighbor Classification

K-nearest neighbor (KNN) is one of the most commonly used methods in classification. Given an unlabeled sample, KNN finds the k nearest neighbors in the reference set based on similarity measures and predicts the label of the sample using the labels of the k nearest neighbors. For example, assume we have an unlabeled sample X and a reference set $D = d_1, d_2, \dots, d_n$, the probability X belonging to class c_j , $P(X, c_j)$, is defined as:

$$P(X, c_j) = \sum_{d_i \in KNN} D(X, d_i)P(d_i, c_j) - b_j \quad (\text{B.1})$$

where $D(X, d_i)$ denotes the similarity between X and d_i and b_j is a bias term. The class label of X is predicted as $\arg_{c_j} \max P(X, c_j)$. Pearson's coefficient correlation and Euclidean distance are two most often used similarity measures.

Due to the natural simplicity, KNN is not robust to data noise and requires better quality of discriminative features. Hence KNN always cooperates with feature selection to fulfill the tasks on machine learning problems.

B.2 Support Vector Machine

The basic idea of support vector machine (SVM) [79] is to find a hyperplane that separates the class with maximum margin. The idea of margin makes SVM less over-fit the training data. Consider one has a training dataset with d points $\{\mathcal{X}_i, \mathcal{Y}_i\}$ $i \in (1, \dots, d)$, where $\mathcal{X}_i \in R^n$ is training sample and $\mathcal{Y}_i \in \{-1, 1\}$ is the class label. A linear classification function is assumed to be:

$$f(\mathcal{X}) = (W \cdot \mathcal{X}) + b \quad (\text{B.2})$$

In binary classification problem, a new sample $\mathcal{X}_{new}$ is predicted as $\mathcal{Y}_{new} = \text{sign}[f(\mathcal{X}_{new})]$ based on the hyperplane $(W \cdot \mathcal{X}) + b = 0$, $W \in R^n$, $b \in R$, where W represents the normal vector of hyperplane and $W \cdot \mathcal{X}$ corresponds to the dot product of the two vectors. The distance from the hyperplane to the nearest points of classes is called the margin and calculated with $1/\|W\|$. To maximize the margin as well as to minimize the misclassification loss, the problem is equivalent to find the solution of the following optimization problem:

$$\begin{aligned} \min_{w, b, \xi} & \left(\frac{1}{2} \|W\|^2 + C \sum_i^d \xi_i \right) \\ \text{subject to } & \forall i, \mathcal{Y}_i f(\mathcal{X}_i) \geq 1 - \xi_i \end{aligned} \quad (\text{B.3})$$

where ξ_i denotes the distance between the misclassified points and the closest corresponding class. Equation B.3 trades off the two goals of finding a hyperplane with large margin (minimizing the first term) and finding a hyperplane that separates the

data well (minimizing the second term). The trade-off is controlled by the parameter C .

For those datasets that are not linearly separable, SVM also provides a variation to construct nonlinear separating surfaces by mapping the data to a higher dimensional feature space, $\mathcal{X} \rightarrow \Phi(\mathcal{X})$. Such that the solution to Equation B.3 becomes $f(\mathcal{X}) = W \cdot \Phi(\mathcal{X}) + b$. We can solve this optimization problem using Lagrangian theory and prove that the vector W is of the form $W = \sum_{i=1}^d \alpha_i^0 Y_i \Phi(\mathcal{X}_i)$, where α_i^0 is the solution of the following quadratic optimization problem:

$$\begin{aligned} \max_{\alpha} W(\alpha) &= \sum_{i=1}^d \alpha_i - \frac{1}{2} \sum_{i,j=1}^d \alpha_i \alpha_j \mathcal{Y}_i \mathcal{Y}_j (K(\mathcal{X}_i, \mathcal{X}_j) + \frac{1}{C} \delta_{i,j}) \\ &\text{subject to } \sum_{i=1}^d \mathcal{Y}_i \alpha_i = 0 \text{ and } \forall i, \alpha_i \geq 0 \end{aligned} \quad (\text{B.4})$$

where $\delta_{i,j}$ denotes the Kronecker symbol and $K(\mathcal{X}_i, \mathcal{X}_j) = \Phi(\mathcal{X}_i) \cdot \Phi(\mathcal{X}_j)$ is the kernel function of the training samples. The most commonly used kernels are the polynomial kernel ($K(x, y) = (x \cdot y + 1)^n$) and Gaussian kernel ($K(x, y) = \exp(-\|x - y\|^2 / 2\sigma^2)$).

B.2.1 Radius Margin Bound

There are many statistics for estimating the generalization performance of SVM from bounds on the LOOCV error L . One of the most used LOOCV error bounds for SVMs is the radius margin bound (only for decision function with non-zero bias b) [79].

Definition B.14 Radius Margin Bound: *If images of training data of size l belonging to a sphere of size R are separable with the corresponding margin γ , then the expectation of the error probability has the bound:*

$$L \leq \frac{1}{d} \frac{R^2}{\gamma^2} \quad (\text{B.5})$$

The radius R of the smallest sphere enclosing the training points can be achieved by solving the following quadratic problem [79]:

$$\begin{aligned} R^2 = \max_{\beta} \sum_{i=1}^d \beta_i K(\mathcal{X}_i, \mathcal{X}_j) - \sum_{i,j=1}^d \beta_i \beta_j K(\mathcal{X}_i, \mathcal{X}_j) \\ \text{subject to } \sum_{i=1}^d \beta_i = 1 \text{ and } \forall i, \beta_i \geq 0 \end{aligned} \quad (\text{B.6})$$

On the other hand, the margin γ is calculated based on $\gamma = 1/\|W\|$ and $\|W\|^2 = 2W(\alpha^0) = 2 \sum_{i,j=1}^d \alpha_i^0 \alpha_j^0 \mathcal{Y}_i \mathcal{Y}_j K(\mathcal{X}_i, \mathcal{X}_j)$, where α^0 is the solution of Equation B.4. In practice, the radius margin bound is calculated as $R^2 \|W\|^2 / d$.

Appendix C

An Example of FCR and PCR Features

Table C.1: A Simple Example of FCR and PCR Features

X_1	X_2	X_3	X_4	C
-1	0	-1	-1	c_1
-1	0	1	1	c_1
0	-1	0	-1	c_2
0	1	0	1	c_2
1	-1	-1	-1	c_3
1	1	1	1	c_3

To further illustrate the meaning of FCR and PCR features, we use a simple example in Table C.1. Note that feature X_1 represents an instance of FCR feature since it is relevant to all SVS sets and is crucial for distinguishing any pair classes. X_2 on the other hand is a PCR feature since it is irrelevant to the SVS set $\{\{c_2\}, \{c_3\}\}$. X_2 is useful only for distinguish between class c_1 and classes $\{c_2, c_3\}$ but not between class c_2 and c_3 . X_3 is also a PCR feature since it is irrelevant to SVS set $\{\{c_1\}, \{c_3\}\}$, while X_4 represents an irrelevant feature. It is worth noting that PCR features are likely to be regarded as irrelevant, especially when the learning sample size of each

class is small. To illustrate this, we evaluate the relevance of all features listed in Table 14 using feature selection algorithms, ReliefF [24], namely, Chi-Square [26], and Gain Ratio [25]. The results obtained indicate that PCR features, X_2 and X_3 , and the irrelevant feature X_4 , have similar low values of relevance. Hence, X_2 and X_3 are likely to be considered as irrelevant features just like X_4 .

Appendix D

Microarray Datasets

Table D.1: Description of ALL Microarray Datasets Used in This Dissertation

Dataset	Genes	Samples	Classes	Description and Reference
ALL-AML	7129	72	2	Two acute leukemia: Acute Myelogenous Leukemia (AML) and Acute Lymphoblastic Leukemia (ALL) [89]
ALL-AML1000	1000	72	2	ALL-AML dataset with 1000 preselected genes [85].
ALL-AML-3	7129	72	3	AML, ALL B-cell, and ALL T-Cell [89].
ALL-AML-4	7129	72	4	AML-Bone Marrow, AML-Peripheral Blood, ALL B-cell, and T-Cell [89].
Colon	2000	62	2	40 colon cancer biopsies vs. 22 normal biopsies [122].
Colon1000	1000	62	2	Colon dataset with 1000 preselected genes [85].
CNS	7129	60	2	Outcome of the treatments for 60 central nervous system cancer patients (21 survivors and 39 are failures) [123].
Breast	24481	97	2	97 samples from breast cancer patients (46 patients developed distance metastases, the rest 51 remained healthy after their initial diagnosis for interval of at least 5 years) [124].
Lung	12533	181	2	Classification between malignant pleural mesothelioma (MPM) and adenocarcinoma (ADCA) of the lung (31 MPM and 150 ADCA) [125].
Lung5c	12600	203	5	4 lung cancer types and 1 normal tissues [119].
Ovarian	15154	253	2	The proteomic spectra of 91 normal persons and 162 ovarian cancer patients [126].
Lymphoma	4026	62	3	Three most prevalent adult lymphoid tumors [113].
MLL	12582	72	3	AML, ALL, and mixed-lineage leukemia (MLL) [114].
SRBCT	2308	83	4	Small, round blue cell tumors of childhood [86].
NCI60	6114	61	9	61 cell lines from 9 human cancer types [115].
NCI1000	1000	61	9	NCI60 dataset with 1000 preselected genes [16, 17].
Brain	10367	50	4	4 malignant glioma types [116].
Thyroid	2000	168	4	3 thyroid tumor types (Follicular adenoma, Follicular carcinoma, and Papillary Carcinoma) and 1 normal tissues [117].
ALL	12558	248	6	6 subtypes of ALL [118].

Bibliography

- [1] H. Liu and L. Yu, “Toward integrating feature selection algorithms for classification and clustering,” *IEEE Transaction on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–501, 2005.
- [2] L. Yu, *Redundancy-based feature selection for high-dimensional data and application in bioinformatics*, Phd Thesis, Arizona State University, 2005.
- [3] P. Moscato, “On evolution, search, optimization, genetic algorithms and martial arts: toward memetic algorithms,” Tech. Rep. Caltech Concurrent Computation Program, Rep. 826, California Inst. Technol., Pasadena, CA, 1989.
- [4] Y. S. Ong and A. J. Keane, “Meta-lamarckian in memetic algorithm,” *IEEE Transaction on Evolutionary Computation*, vol. 8, no. 2, pp. 99–110, 2004.
- [5] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, “Classification of adaptive memetic algorithms: A comparative study,” *IEEE Transactions On Systems, Man and Cybernetics - Part B*, vol. 36, no. 1, pp. 141–152, 2006.
- [6] T. Yoshida H. Ishibuchi and T. Murata, “Balance between genetic search and local search in memetic algorithm for multiobjective permutation flowshop

- scheduling,” *IEEE Transaction on Evolutionary Computation*, vol. 7, no. 2, pp. 204–223, 2003.
- [7] N. Krasnogor, *Studies on the Theory and Design Space of Memetic Algorithms*, Ph.D. Thesis, Faculty of Computing, Mathematics and Engineering, University of the West of England, Bristol, U.K, 2002.
- [8] M. Dash and H. Liu, “Feature selection for classification,” *Intelligent Data Analysis: An International Journal*, vol. 1, no. 3, pp. 131–156, 1997.
- [9] M. Dash and H. Liu, “Consistency-based search in feature selection,” *Artificial Intelligence*, vol. 151, no. 1-2, pp. 155–176, 2003.
- [10] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Boston: Kluwer Academic, 1998.
- [11] R. Kohavi and G. H. John, “Wrapper for feature subset selection,” *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.
- [12] J. H. Holland, *Adaptation in natural artificial systems, 2nd edition*, MIT Press, 1992.
- [13] L. Li, L. G. Pedersen, T. A. Darden, and C. R. Weinberg, “Computational analysis of leukemia microarray expression data using ga/knn method,” *Proceeding of the 1st Conference on Critical Assessment of Microarray Data Analysis, CAMDA2000*, 2000.

- [14] L. Li, C. R. Weinberg, T. A. Darden, and L. G. Pedersen, "Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the ga/knn method," *Bioinformatics*, vol. 17, no. 12, pp. 1131–1142, 2001.
- [15] T. Jirapech-Umpai and S. Aitken, "Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes," *BMC Bioinformatics*, vol. 6, no. 148, 2005.
- [16] C. H. Ooi and P. Tan, "Genetic algorithms applied to multi-class prediction for the analysis of gene expression data," *Bioinformatics*, vol. 19, no. 1, pp. 37–44, 2003.
- [17] J. J. Liu, G. Cutler, W. Li, Z. Pan, S. Peng, T. Hoey, L. Chen, and X. B. Ling, "Multiclass cancer classification and biomarker discovery using ga-based algorithms," *Bioinformatics*, vol. 21, no. 11, pp. 2691–2697, 2005.
- [18] L. Li, W. Jiang, X. Li, K. L. Moser, Z. Guo, L. Du, Q. Wang, E. J. Topol, Q. Wang, and S. Rao, "A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset," *Genomics*, vol. 85, pp. 16–23, 2005.
- [19] J. M. Deutsch, "Evolutionary algorithms for finding optimal gene sets in microarray prediction," *Bioinformatics*, vol. 19, no. 1, pp. 45–52, 2003.
- [20] J. H. Yang and V. Honavar, "Feature selection using a genetic algorithm," *IEEE Transaction on Intelligent Systems*, vol. 13, no. 2, pp. 44–49, 1998.

- [21] M. L. Raymer, W. F. Punch, E. D. Goodman, L. A. Kuhn, and A. K. Jain, "Dimensionality reduction using genetic algorithms," *IEEE Transaction on Evolutionary Computation*, vol. 4, no. 2, pp. 164–171, 2000.
- [22] Z. Zhu, Y. S. Ong, and M. Dash, "Wrapper-filter feature selection algorithm using a memetic framework," *IEEE Transactions On Systems, Man and Cybernetics - Part B*, vol. 37, no. 1, pp. 70–76, 2007.
- [23] D.J. Newman, S. Hettich, C.L. Blake, and C.J. Merz, "UCI repository of machine learning databases," University of California, Irvine, Department of Information and Computer Sciences, <http://www.ics.uci.edu/~mlearn/MLRepository.html>, 1998.
- [24] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of relieff and rrelieff," *Machine Learning*, vol. 53, no. 1-2, pp. 23–69, 2003.
- [25] J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo, Morgan Kaufman, 1993.
- [26] H. Liu and R. Setiono, "Chi2: Feature selection and discretization of numeric attributes," In Proceedings of 7th IEEE International Conference on Tools with Artificial Intelligence, 1995.
- [27] D. Koller and M. Sahami, "Toward optimal feature selection," Proceedings of the 13th International Conference on Machine Learning, pp. 284–292, 1996.

- [28] L. Yu and H. Liu, “Efficient feature selection via analysis of relevance and redundancy,” *Journal of Machine Learning Research*, vol. 5, pp. 1205–1224, 2004.
- [29] Z. Zhu, Y. S. Ong, and M. Dash, “Markov blanket-embedded genetic algorithm for gene selection,” *Pattern Recognition*, 2007(In Press).
- [30] Z. Zhu, Y. S. Ong, and E. P. Lim, “Simultaneous identification of full class relevant and partial class relevant genes,” *BMC Bioinformatics*, 2007(Submitted).
- [31] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, San Mateo: Morgan Kaufmann, 1988.
- [32] M. A. Hall, “Correlation-based feature selection for discrete and numeric class machine learning,” *Proceedings of the 17th International Conference on Machine Learning*, pp. 359-366, 2000.
- [33] H. Almuallim and T. G. Dietterich, “Learning boolean concepts in the presence of many irrelevant features,” *Artificial Intelligence*, vol. 69, no. 1-2, pp. 279–305, 1994.
- [34] J. R. Quinlan, “Introduction of decision trees,” *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.
- [35] E. Xing, M. Jordan, and R. Karp, “Feature selection for high-dimensional genomic microarray data,” In *Proceedings of 15th International Conference on Machine Learning*, 2001.

- [36] S. Das, “Filters, wrappers and a boosting-based hybrid for feature selection,” In Proceedings of 18th International Conference on Machine Learning, 2001.
- [37] P. Narendra and K. Fukunaga, “A branch and bound algorithm for feature subset selection,” *IEEE Transaction on Computers*, vol. 26, no. 9, pp. 917–922, 1977.
- [38] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Pentice-Hall International, 1982.
- [39] A. J. Miller, *Subset Selection in Regression*, Chapman and Hall, 1990.
- [40] P. Pudil, J. Novovicova, and J. Kittler, “Floating search methods in feature selection,” *Pattern Recognition Letters*, vol. 15, pp. 1119–1125, 1994.
- [41] H. Liu and R. Setiono, “A probabilistic approach to feature selection - a filter solution,” Proceedings of 13th International Conference on Machine Learning, pp. 319-327, 1996.
- [42] G. Brassard and P. Bratley, *Fundamentals of Algorithms*, New Jersey: Prentice Hall, 1996.
- [43] Z. X. Zhu, Y. S. Ong, K. W. Wong, and K. T. Seow, “Experimental condition selection in whole-genome functional classification,” IEEE Conference on Cybernetic and Intelligent System, Singapore, December, 2004.

- [44] I. S. Oh, J. S. Lee, and B. R. Moon, “Hybrid genetic algorithm for feature selection,” *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1424–1437, 2004.
- [45] M. Wahde and Z. Szallasi, “A survey of methods for classification of gene expression data using evolutionary algorithms,” *Expert Review of Molecular Diagnostic*, vol. 6, no. 1, pp. 101–110, 2006.
- [46] H. Ishibuchi and T. Nakashima, “Multi-objective pattern and feature selection by a genetic algorithm,” In Proc. of Genetic and Evolutionary Computation Conference (Las Vegas, Nevada, U.S.A.) pp.1069-1076, 2000.
- [47] J. Liu and H. Iba, “Selecting informative genes using a multiobjective evolutionary algorithm,” In Proc 2002 Congress on Evolutionary Computation, 2002.
- [48] P. Moscato, *Memetic Algorithm: A Short Introduction*, In New Ideas in Optimization, ch. 14, pp. 219-234, McGraw-Hill, London, 1999.
- [49] M. W. S. Land, *Evolutionary Algorithms with Local Search for Combinatorial Optimization*, Phd Thesis, University of California, San Diego, 1998.
- [50] D. P. Merz, *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies*, Ph.D. Thesis, Department of Electrical Engineering and Computer Science, University of Siegen, 2000.
- [51] P. Merz, “On the performance of memetic algorithms in combinatorial optimization,” GECCO 2001, San Francisco, CA, USA, 2001.

- [52] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, Berlin: Springer, 1996.
- [53] D. E. Goldberg and S. Voessner, “Optimizing global-local search hybrids,” In: Proc. 1999 Genetic and Evolutionary Computation Conf., pp. 220-228, 1999.
- [54] W. E. Hart, *Adaptive Global Optimization with Local Search*, Ph.D. Thesis, Department of Computer Science and Engineering, University of California, San Diego, 1994.
- [55] C. Guerra-Salcedo, S. Chen, D. Whitley, and S. Smith, “Fast and accurate feature selection using hybrid genetic strategies,” In: Proc. 1999 Genetic and Evolutionary Computation Conf., pp. 177-184, 1999.
- [56] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Reading, Mass., Addison-Wesley, 1989.
- [57] K. Deb and A. R. Reddy, “Reliable classification of two-class cancer data using evolutionary algorithms,” *BioSystems*, vol. 72, pp. 111–129, 2003.
- [58] M. Banerjee, S. Mitra, and A. Anand, *Feature Selection Using Rough Sets*, chapter 1, Yaochu Jin (editor), Multi-Objective Machine Learning. Springer, Berlin Heidelberg, 2006.
- [59] J. Handl and J. Knowles, *Multi-Objective Clustering and Cluster Validation*, chapter 2, Yaochu Jin (editor), Multi-Objective Machine Learning. Springer, Berlin Heidelberg, 2006.

- [60] M. Morita, R. Sabourin, F. Bortolozzi, and C. Suen, “Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition,” In Proc. 7th International Conference on Document Analysis and Recognition (ICDAR2003), 2003.
- [61] H. Ishibuchi and T. Murata, “Multi-objective genetic local search algorithm,” Proceedings of 1996 IEEE International Conference on Evolutionary Computation (ICEC96), May 20–22, Nagoya University, Japan, Piscataway, NJ IEEE Service Center, pp. 119–124, 1996.
- [62] H. Ishibuchi and T. Murata, “A multi-objective genetic local search algorithm and its application to flowshop scheduling,” *IEEE Trans. on Systems, Man, and Cybernetics -Part C: Applications and Reviews*, vol. 28, no. 3, pp. 392–403, 1998.
- [63] H. Ishibuchi and S. Kaige, “Implementation of simple multiobjective memetic algorithms and its application to knapsack problems,” *International Journal of Hybrid Intelligent Systems*, vol. 1, no. 1, pp. 22–35, 2004.
- [64] A. Jaszkiwicz, “Genetic local search for multi-objective combinatorial optimization,” *European Journal of Operational Research*, vol. 137, pp. 50–71, 2002.
- [65] A. Jaszkiwicz, “Do multiple-objective metaheuristics deliver on their promise? a computational experiment on the set-covering problem,” *IEEE Transaction on Evolutionary Computation*, vol. 7, no. 2, pp. 133–143, 2003.

- [66] J. D. Knowles and D. W. Corne, “M-paes: A memetic algorithm for multiobjective optimization,” *Proceedings of Congress of Evolutionary Computation CEC2000*, 2000.
- [67] B. Efron and R. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, 1993.
- [68] M. Chernick, *Bootstrap Methods: A Practitioner’s Guide*, Wiley, New York, NY, 1999.
- [69] U. M. Braga-Neto and E. R. Dougherty, “Is cross-validation valid for small-sample microarray classification?,” *Bioinformatics*, vol. 20, no. 3, pp. 374–380, 2004.
- [70] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, 1995, pp. 1137–1145.
- [71] R. Kohavi, “Feature subset selection using the wrapper method: overfitting and dynamic search space topology,” in *Proceedings of the First International Conference on Knowledge Discovery and Data Mining*, 1995, pp. 192–197.
- [72] A. Jain and D. Zongker, “Feature selection: evaluation, application, and small sample performance,” *IEEE Transactions and Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.
- [73] J. Reunanen, “Overfitting in making comparisons between variable selection methods,” *Journal of Machine Learning Research*, vol. 3, pp. 1371–1382, 2003.

- [74] C. Ambroise and G. J. McLachlan, “Selection bias in gene extraction on the basis of microarray gene-expression data,” *PNAS*, vol. 99, no. 10, pp. 6562–6566, 2002.
- [75] H. Liu, E. R. Dougherty, J. G. Dy, K. Torkkola, E. Tuv, H. Peng, C. Ding, F. Long, M. Berens, L. Parsons, Z. Zhao, L. Yu, and G. Forman, “Evolving feature selection,” *IEEE Intelligent Systems*, vol. 20, no. 6, pp. 64–76, 2005.
- [76] A. Statnikov, C. F. Aliferis, I. Tsamardinos, D. Hardin, and S. Levy, “A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis,” *Bioinformatics*, vol. 21, no. 5, pp. 631–643, 2005.
- [77] S. K. Singhi and H. Liu, “Feature subset selection bias for classification learning,” *ICML ’06: Proceedings of the 23rd international conference on Machine learning*, 2006.
- [78] C.F. Aliferis, A. Statnikov, and I. Tsamardinos, “Challenges in the analysis of mass-throughput data: A technical commentary from the statistical machine learning perspective,” *Cancer Informatics*, vol. 2, pp. 133–162, 2006.
- [79] V. Vapnik, *Statistical Learning Theory*, John Wiley, 1998.
- [80] J. Loughrey and P. Cunningham, “Using early-stopping to avoid overfitting in wrapper-based feature selection employing stochastic search,” Technical Report TCD-CS-2005-37, Department of Computer Science, The University of Dublin, Trinity College, 2005.

- [81] G. Schwarzer and W. Vach, “The misuses of artificial neural networks for prognostic and diagnostic classification in oncology,” *Stat Med*, vol. 19, no. 4, pp. 541–561, 2000.
- [82] E. E. Ntzani and J. P. Ioannidis, “Predictive ability of dna microarrays for cancer outcomes and correlates: an empirical assessment,” *Lancet*, vol. 362, no. 9394, pp. 1439–1444, 2003.
- [83] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, “Quantitative monitoring of gene expression patterns with a complementary dna microarray,” *Science*, vol. 270, no. 5235, pp. 467C470, 1995.
- [84] J. E. Baker, “Adaptive selection methods for genetic algorithms,” in *Proceedings of the 1st International Conference on Genetic Algorithms*, 1985, pp. 101–111.
- [85] X. Zhou and K. Z. Mao, “Ls bound based gene selection for dna microarray data,” *Bioinformatics*, vol. 21, no. 8, pp. 1559–1564, 2005.
- [86] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, “Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks.,” *Nat Med*, vol. 7, no. 6, pp. 673–679, 2001.

- [87] T. Li, C. Zhang, and M. Ogihara, “A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression,” *Bioinformatics*, vol. 20, no. 15, pp. 2429–2437, 2004.
- [88] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, “Gene selection for cancer classification using support vector machines,” *Machine Learning*, vol. 46, no. 1-3, pp. 389–422, 2002.
- [89] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, “Molecular classification of cancer: class discovery and class prediction by gene expression monitoring,” *Science*, vol. 286, no. 5439, pp. 531–537, 1999.
- [90] L. Yu and H. Liu, “Redundancy based feature selection for microarray data,” in *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 737–742.
- [91] S. Dudoit, J. Fridlyand, and T. P. Speed, “Comparison of discrimination methods for the classification of tumors using gene expression data,” *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.
- [92] H. Liu, J. Li, and L. Wong, “A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns,” *Genome Information*, vol. 13, pp. 51–60, 2002.

- [93] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C*, Cambridge University Press, Cambridge, 1998.
- [94] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, “Incremental wrapper-based gene selection from microarray data for cancer classification,” *Pattern Recognition*, vol. 39, no. 12, pp. 2383–2392, 2006.
- [95] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques, 2nd Edition*, Morgan Kaufmann, San Francisco, 2005.
- [96] G. H. John, R. Kohavi, and K. Pfleger, “Irrelevant features and the subset selection problem,” in *Proceeding of the 11th International Conference on Machine Learning*, 1994, pp. 121–129.
- [97] J. Li and H. Liu, “Kent ridge biomedical data set repository,” [Http://sdmc-lit.org.sg/GEDatasets](http://sdmc-lit.org.sg/GEDatasets), 2002.
- [98] S. Mukherjee, *Classifying Microarray Data Using Support Vector Machines*, In: A Practical Approach to Microarray Data Analysis, D.P. Berrar, W. Dubitzky and M. Granzow (Eds.), Kluwer Academic Publishers, Boston, MA, Chapter 9, 166-185, 2003.
- [99] U. Fayyad and K. Irani, “Multi-interval discretization of continuous-valued at-tributes for classification learning,” In *Proceeding of 10th International Conference of Machine Learning*, pp. 194-201, 1993.

- [100] M. Berens, H. Liu, and L Yu, “Fostering biological relevance in feature selection for microarray data,” *IEEE Transaction on Intelligent Systems*, vol. 20, no. 6, pp. 71–73, 2005.
- [101] A. Rakotomamonjy, “Variable selection using svm based criteria,” *Journal of Machine Learning Research*, vol. 3, pp. 1357–1370, 2003.
- [102] D. V. Nguyen and D. M. Rocke, “Tumor classification by partial least squares using microarray gene expression data,” *Bioinformatics*, vol. 18, no. 1, pp. 39–50, 2002.
- [103] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub, “Multiclass cancer diagnosis using tumor gene expression signatures,” *PNAS*, vol. 98, no. 26, pp. 15149–15154, 2001.
- [104] D. V. Nguyen and D. M. Rocke, “Multi-class cancer classification via partial least squares with gene expression profiles,” *Bioinformatics*, vol. 18, no. 9, pp. 1216–1226, 2002.
- [105] C. H. Ooi, M. Chetty, and S. W. Teng, “Differential prioritization between relevance and redundancy in correlation-based feature selection techniques for multiclass gene expression data,” *BMC Bioinformatics*, vol. 7, no. 320, 2006.

- [106] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu, “Diagnosis of multiple cancer types by shrunken centroids of gene expression,” *PNAS*, vol. 99, no. 10, pp. 6567–6572, 2002.
- [107] K. Y. Yeung and R. E. Bumgarner, “Multiclass classification of microarray data with repeated measurements: application to cancer,” *Genome Biology*, vol. 4, no. 12, pp. R83, 2003.
- [108] K. Y. Yeung, R. E. Bumgarner, and A. E. Raftery, “Bayesian model averaging: development of an improved multi-class, gene selection and classification tool for microarray data,” *Bioinformatics*, vol. 21, no. 10, pp. 2394–2402, 2005.
- [109] L. S. Oliveira, M. Morita, and R. Sabourin, *Feature Selection for Ensembles Using the Multi-Objective Optimization Approach*, chapter 3, Yaochu Jin (editor), Multi-Objective Machine Learning. Springer, Berlin Heidelberg, 2006.
- [110] A. Tsymbal, S. Puuronen, and D. W. Patterson, “Ensemble feature selection with the simple bayesian classification,” *Information Fusion*, vol. 4, no. 2, pp. 87–100, 2003.
- [111] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transaction on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.
- [112] R. Diaz-Uriarte and S. A. de Andres, “Gene selection and classification of microarray data using random forest,” *BMC Bioinformatics*, vol. 7, no. 3, 2006.

- [113] A. A. Alizadeh, M. B. Eisen, and E. E. Davis et al., “Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling,” *Nature*, vol. 403, no. 6769, pp. 503–511, 2000.
- [114] S. A. Armstrong, J. E. Staunton, L. B. Silverman, R. Pieters, M. L. den Boer, M. D. Minden, S. E. Sallan, E. S. Lander, T. R. Golub, and S. J. Korsmeyer, “Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia,” *Nature Genetics*, vol. 30, no. 1, pp. 41–47, 2002.
- [115] D. T. Ross, U. Scherf, and M. B. Eisen et al., “Systematic variation in gene expression patterns in human cancer cell lines,” *Nature Genetics*, vol. 24, no. 3, pp. 208–209, 2000.
- [116] C. L. Nutt, D. R. Mani, and R. A. Betensky et al., “Gene expression-based classification of malignant gliomas correlates better with survival than histological classification,” *Cancer Research*, vol. 63, no. 7, pp. 1602–1607, 2003.
- [117] N. Yukinawa, S. Oba, K. Kato, K. Taniguchi, K. Iwao-Koizumi, Y. Tamaki, S. Noguchi, and S. Ishii, “A multi-class predictor based on a probabilistic model: application to gene expression profiling-based diagnosis of thyroid tumors,” *BMC Genomics*, vol. 7, no. 190, 2006.
- [118] E. J. Yeoh, M. E. Ross, and S. A. Shurtleff et al., “Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling,” *Cancer Cell*, vol. 1, no. 2, pp. 109–110, 2002.

- [119] A. Bhattacharjee, W. G. Richards, and J. Stauton et al., “Classification of human lung carcinomas by mrna expression profiling reveals distinct adenocarcinoma subclasses,” *PNAS*, vol. 98, no. 24, pp. 13790–13795, 2001.
- [120] S. R. Bauer, H. Kubagawa, I. MacLennan, and F. Melchers, “Vpreb gene expression in hematopoietic malignancies: a lineage- and stage-restricted marker for b-cell precursor leukemias,” *Blood*, vol. 78, no. 6, pp. 1581–1588, 1991.
- [121] A. V. Krivtsov, D. Twomey, Z. Feng, M. C. Stubbs, Y. Wang, J. Faber, J. E. Levine, J. Wang, W. C. Hahn, D. G. Gilliland, and T. R. Golub and S. A. Armstrong, “Transformation from committed progenitor to leukemia stem cell initiated by mll-af9,” *Nature*, vol. 442, no. 7104, pp. 818–822, 2006.
- [122] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays,” *PNAS*, vol. 96, no. 12, pp. 6745–6750, 1999.
- [123] S. L. Pomeroy, P. Tamayo, and M. Gaasenbeek et al., “Prediction of central nervous system embryonal tumour outcome based on gene expression,” *Nature*, vol. 415, no. 6870, pp. 436–442, 2002.
- [124] L. J. Van’t Veer, H. Dai, and M. J. van de Vijver et al., “Gene expression profiling predicts clinical outcome of breast cancer,” *Nature*, vol. 415, no. 6871, pp. 530–536, 2002.

- [125] G. J. Gordon, R. V. Jensen, and L. Hsiao et al., “Translation of microarray data into clinically relevant cancer diagnostic tests using gene expression ratios in lung cancer and mesothelioma,” *Cancer Research*, vol. 62, pp. 4963–4967, 2002.

- [126] E. F. Petricoin, A. M. Ardekani, and B. A. Hitt et al., “Use of proteomic patterns in serum to identify ovarian cancer,” *Lancet*, vol. 359, no. 9306, pp. 572–577, 2002.