STABLE FEATURE SELECTION: THEORY AND ALGORITHMS

BY

YUE HAN

B.Eng, Beijing Jiaotong University, 2007

DISSERTATION

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Computer Science
in the Graduate School of
Binghamton University
State University of New York
2012

Accepted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Computer Science
in the Graduate School of
Binghamton University
State University of New York
2012

Dr. Lei Yu _____ April 26, 2012
(Dissertation Advisor)
Department of Computer Science

Dr. Weiyi Meng _____ April 26, 2012
Department of Computer Science

Dr. Michal Cutler _____ April 26, 2012
Department of Computer Science

Dr. Xingye Qiao _____ April 26, 2012
Department of Mathematical Sciences

Dr. Harold Lewis _____ April 26, 2012
(External Member)
Department of Systems Science and Industrial Engineering

Abstract

Feature selection plays an important role in knowledge discovery from many application domains with high-dimensional data. Many feature selection algorithms have been developed and shown successful at improving predictive accuracy of learning models while reducing feature space dimensionality and model complexity. Besides high accuracy, the stability of feature selection - the insensitivity of the result of a feature selection algorithm to variations to the training set, is another important yet under-addressed issue for feature selection. The stability issue has become increasingly critical in application domains where feature selection is used as a knowledge discovery tool to identify important features for explaining the observed phenomena, such as biomarker identification in cancer diagnosis.

In this dissertation, we present a theoretical framework about the relationship between the stability and the accuracy of feature selection based on a formal bias-variance decomposition of feature selection error. The framework also reveals the connection between stability and sample size and suggests a variance reduction approach for improving the stability of feature selection algorithms under small sample size. Following the theoretical framework, we also develop an empirical variance reduction framework and margin based instance weighting algorithms under this framework. Moreover, our extensive experimental study verifies the theoretical framework and the empirical framework based on both synthetic data sets and real-world microarray data sets. Our results show that the empirical framework is effective at reducing the variance and improving the stability of two representative feature selection algorithms, SVM-RFE and ReliefF, while maintaining comparable predictive accuracy based on the selected features. The proposed instance weighting framework is also shown to be more effective and efficient than the existing ensemble framework at improving the subset stability of the feature selection algorithms under study.

For my parents, Wei Han and Xianning Liu

without whom none of this would have mattered

Acknowledgments

It is my great pleasure to give my sincere thanks to all the people who have helped me during the course of my studies in Binghamton University for the past five years.

First of all I would like to express my deepest gratitude towards my advisor, Dr. Lei Yu, Associate professor in Department of Computer Science, Binghamton University. Without his guidance and support extended to me during my Ph.D study, I would not have made such progress towards the completion of my degree. Without his thorough supervision, the work would have never taken shape. I humbly present my wholehearted thanks to him.

Secondly, I would like to thank the committee members of my dissertation defense, Dr. Weiyi Meng and Dr. Michal Cutler from Department of Computer Science, Dr. Xingye Qiao from Department of Mathematics and Dr. Harold Lewis from Department of Systems Science and Industrial Engineering at Binghamton University. Their comments and suggestions during the proposal of my dissertation shed lights on my further study and exploration of this topic.

Last but not least, I would like to express my thanks to my labmates Dr. Jian Wang, Ruiqi Luo, Steven Loscalzo, Yi Xu and other classmates for their great help during my studies.

Once again I simply and humbly say thanks to you all.

Contents

List of Tables

List of Figures

List of Abbreviations

AUC	Area Under Curve
CV	Cross Validation
En	Ensemble
FS	Feature Selection
HHSVM	Hybrid Hyberized Support Vector Machines
HM	Hypothesis Margin
IW	Instance Weighting
LLE	Locally Linear Embedding
MVU	Maximum Variance Unfolding
mrMR	Minimum Redundancy and Maximum Relevance
1NN	One Nearest Neighbor
POGR	Percentage of Overlapping Genes Related
PCA	Principle Component Analysis
ROC	Receiver Operating Characteristic
SM	Sample Margin
SVM	Support Vector Machine
SVM-RFE	SVM-Recursive Feature Elimination

1 Introduction

The data dimensionality grows rapidly in a broad spectrum of application domains, which brings great challenges to traditional data mining and machine learning tasks in terms of scalability, effectiveness and speed. Without modifying existing approaches, feature selection can alleviate the problem of high dimensionality. Accordingly it has been well recognized as a key step during the process of knowledge discovery and has attracted growing interest from both the academia and the industry in the past. By removing irrelevant and redundant features based on certain feature evaluation measures, feature selection helps enhance the generalization capability of learning models, speed up the learning process and improve the model interpretability. A great variety of feature selection algorithms have been developed with a focus on improving the predictive accuracy of learning models while reducing dimensionality and model complexity.

Although it is important to evaluate the performance of a feature selection algorithm based on the predictive accuracy of models built on the selected features, researchers are drawing attention to a more comprehensive evaluation of feature selection algorithms besides the predictive accuracy. *Stability of feature selection*, the insensitivity of the result of a feature selection algorithm to variations to the training set, is emerging as another important aspect under study. This issue is particularly critical for application domains where feature selection is used as a knowledge dis-

covery tool for identifying robust underlying characteristic features which remain the same through any training data variation. A good feature selection algorithm should not only help achieve better prediction performance but also produce stable selection results as training data varies.

In this introductory chapter, we point out the new challenges for feature selection on high-dimensional data in Section 1.1 and further motivate the need of stable feature selection under these challenges in Section 1.2. Key factors which contribute to the instability of feature selection are explained in Section 1.3. In Section 1.4, we summarize our major contributions in this dissertation.

1.1 High-dimensional Data and Feature Selection

Nowadays, people are overwhelmed by the huge amount of data while it is still accumulating in a speed unreachable by human's capability of processing the data. Database technology has been used with great success in the collection and organization of prosperous data, which forms a potential gold mine of valuable information for different uses. Therefore, data mining as a multidisciplinary joint effort from databases, machine learning, and statistics, is championing in turning mountains of data into nuggets [Agrawal et al. 1993; Devijver and Kittler 1982; Liu and Yu 2005].

In many scientific and application domains, the introduction of delicate and complicated technologies allow people to perform deeper and broader investigation on those datasets with growing dimensionality. For example, high-throughput genomic and proteinic technologies are widely used in cancer research to build better predictive models of diagnosis, prognosis and therapy, to identify and characterize key signalling networks and to find new targets for drug development [Clarke et al. 2008]; a combination of information retrieval and high-dimensional learning schemes in modern text classification systems can be better adopted for automated population of hierarchical catalogues of Web resources, spam filtering, identification of document genre and even

automated essay grading [Sebastiani 2005]. However, most of the methods designed for use in application domains where data usually have a relatively low dimension are not applicable to problems dealing with high-dimensional data. It emerges as the curse of dimensionality when traditional schemes are challenged by the increasing dimensionality of datasets in terms of scalability, effectiveness and speed.

In order to reduce the complexity of data mining problems and adapt existing machine learning approaches to the new scenarios where features are presented in an unprecedented large scale without losing their effectiveness, data preprocessing established itself as a key step during the whole process of knowledge discovery [Liu and Motoda 2001; Pyle 1999]. Among various preprocessing techniques, feature selection is widely used due to its good theoretical and empirical properties [Blum and Langley 1997; Liu and Motoda 2001]. Feature selection, the process of selecting a subset of relevant features for building robust learning models, has attracted growing interest from both the academia and the industry in past decade. By removing irrelevant and redundant features from the data based on certain feature evaluation measures, feature selection helps alleviate the negative effect of the curse of dimensionality, enhance the generalization capability of learning models, speed up the learning process and improve the model interpretability.

Feature selection has been a fertile field of research and development since the 1970s in statistical pattern recognition, machine learning, and data mining, and widely applied to many fields such as text categorization [Forman 2003], image retrieval [Mitchell et al. 2004], customer relationship management, intrusion detection, and genomic analysis [Liu and Yu 2005; Golub et al. 1999]. A great variety of feature selection algorithms have been developed with a focus on improving classification accuracy while reducing dimensionality and model complexity.

1.2 Importance of Stable Feature Selection

The identification and validation of molecular biomarkers for cancer diagnosis, prognosis, and therapeutic targets is an important problem in cancer genomics. Due to the time-consuming, costly, and labor-intensive nature of clinical and biological validation experiments, it is crucial to select a list of high-potential biomarker candidates for validation [Pepe et al. 2001]. Gene expression microarray data [Golub et al. 1999] and comparative genomic hybridization (CGH) microarrays [Pinkel et al. 1998] are widely used for identifying candidate genes in various cancer studies. From a machine learning viewpoint, the selection of candidate genes in this context can be regarded as a problem of feature selection from high-dimensional labeled data, where the aim is to find a small subset of features (genes) that best explains the difference between samples of distinct phenotypes. There exists various feature selection algorithms well adapted into problem of gene marker selection and they aim at improving classification accuracy while reducing dimensionality and model complexity [Li et al. 2004; Liu and Yu 2005; Wasikowski and Chen 2010]. Traditionally, the relevance of a feature is the most important selection criterion because using highly relevant features improves generalization accuracy and avoids the effect of overfitting [John et al. 1994]. A majority of feature selection algorithms concentrate on feature relevance [Liu and Yu 2005]. In order to have better generalization ability, the selected features need to cover a broader space in the feature space, which requires the selected features to be non-redundant or have small overlap. There are several studies on feature redundancy [Ding and Peng 2003; Yu and Liu 2004], which consider the trade-off between feature relevance and redundancy.

Besides high accuracy, another important issue is *stability of feature selection* - the insensitivity of the result of a feature selection algorithm to variations to the training set. This issue is particularly critical for applications where feature selection is used

as a knowledge discovery tool for identifying robust and truly relevant underlying characteristic features which stay the same as training data varies.

Let us continue the example from the domain of biological analysis we introduced earlier. Characteristic markers are a set of stable genes which are used to explain the observed symptoms and disease types. Many feature selection methods have been adopted for gene selection from microarray data, and have shown good classification performance of the selected genes. However, a common problem with existing gene selection methods is that the selected genes by the same method often vary significantly with some variations of the samples in the same training set. To make the matters worse, different methods or different parameter settings of the same method may also result in largely different subsets of genes (gene signatures) for the same set of samples. The instability of the gene signatures raises serious doubts about the reliability of the selected genes as biomarker candidates and hinders biologists from deciding candidates for subsequent validations.

In fact, biologists are interested in finding a small number of features (genes or proteins) that explain the mechanisms driving different behaviors of microarray samples [Pepe et al. 2001]. Biologists instinctively have high confidence in the result of an algorithm that selects similar sets of genes under some variations to the microarray samples. Although most of these subsets are as good as each other in terms of predictive performance [Davis et al. 2006; Kalousis et al. 2007; Loscalzo et al. 2009], such instability dampens the confidence of domain experts in experimentally validating the selected features.

1.3 Factors for Instability of Feature Selection

By observing the characteristics of datasets from different application domains, especially in bioinformatics, a key reason for instability of feature selection on high-dimensional data is the relatively small number of samples(instances). Analysis of

data sets with high dimensionality and low sample size is also noted as "strip mining" [Kewley et al. 2000]. Suppose we perform feature selection on a dataset D with n samples and p features. If we randomly split the data into two sets D_1 and D_2 with half of the samples each, and run a feature selection algorithm on them, ideally, we would like to see the same feature selection result. However, in reality, due to the limited sample size of D, the results from D_1 and D_2 normally do not agree with each other. When the sample size of D is small, the results of feature selection may be even largely different. For microarray data, the typical number of features (genes) is thousands or tens of thousands, but the number of samples is often less than a hundred. For the same feature selection algorithm, a different subset of features may be selected each time with a slight variation in the training data, which indicates the high sensitivity of a feature selection algorithm to data variations.

The stability of feature selection is a complicated issue. Besides sample size, recent studies on this issue [Kalousis et al. 2007; Loscalzo et al. 2009] have shown that the stability of feature selection results depends on various factors such as data distribution, mechanism of feature selection and so on. Moreover, the stability of feature selection results should be investigated together with the predictive performance of machine learning models built on the selected features. Domain experts will not be interested in a strategy (e.g., arbitrarily selecting the same subset of features regardless of the input samples) that yields very stable feature subsets but bad predictive models. Currently, there exist no theoretical studies on why and how various factors affect the stability of feature selection or the relationship between the stability and predictive performance of feature selection.

1.4 A Summary of Major Contributions

In this study, we present a theoretical framework about feature selection stability based on a formal bias-variance decomposition of feature selection error. The theo-

retical framework explains the relationship between the stability and the accuracy of feature selection, and guides the development of stable feature selection algorithms. It suggests that one does not have to sacrifice predictive accuracy in order to get more stable feature selection results. A better tradeoff between the bias and the variance of feature selection can lead to more stable results while maintaining or even improving predictive accuracy based on the selected features. The framework also reveals the dependency of feature selection stability on the number of instances in a training set (or sample size), and suggests a variance reduction approach for improving the stability of feature selection algorithms under small sample size.

Furthermore, we propose an empirical variance reduction framework - margin based instance weighting. The framework first weights each instance in a training set according to its importance to feature evaluation, and then provides the weighted training set to a feature selection algorithm. The idea of instance weighting is motivated by the theory of importance sampling for variance reduction. Intuitively, instance weighting aims to assign higher weights to instances from regions which contribute more to the aggregate feature weights and assign lower weights to instances from other less important (or outlying) regions. To this end, we introduce a novel concept, margin vector feature space, which enables the the estimation of the importance of instances with respect to (w.r.t.) feature evaluation.

The proposed theoretical and empirical frameworks are validated through an extensive set of experiments. Experiments on synthetic data sets demonstrate the bias-variance decomposition of feature selection error and the effects of sample size on feature selection, using the SVM-RFE algorithm. These experiments also verify the effectiveness of the proposed instance weighting framework at reducing the variance of feature weighting by SVM-RFE, and in turn improving the stability and the predictive accuracy of the selected features by SVM-RFE. Experiments on real-world

microarray data sets further verify that the instance weighting framework is effective at reducing the variance of feature weighting and improving the subset stability for two representative feature selection algorithms, SVM-RFE and ReliefF, while maintaining comparable predictive accuracy based on the selected features. Moreover, the instance weighting framework is shown to be more effective and efficient than a recently proposed ensemble framework for stable feature selection.

The rest of this dissertation is organized as follows. Chapter 2 reviews the background and related work. Chapter 3 introduces our theoretical framework on feature selection stability. Chapter 4 proposes an empirical framework of margin based instance weighting and an efficient algorithm developed under this framework. Chapter 5 describes the general setup of our experimental study on stable feature selection. Chapter 6 evaluates the theoretical and empirical frameworks based on synthetic data. Chapter 7 presents a comprehensive evaluation from the real-world microarray data. Chapter 8 concludes the dissertation and outlines future research directions.

2 Background and Related Work

This chapter gives a more detailed view of feature selection and a comprehensive review of the literature related to our work. Section 2.1 examines the broad categories of feature selection models and several state-of-the-art feature selection algorithms. Section 2.2 begins with the stability of learning algorithms and reviews the existing work on the stability of feature selection with a focus on two recently proposed schemes: Ensemble Feature Selection and Group-based Feature Selection. Section 2.3 explains the concept of margin and describe several representative margin-based feature selection algorithms. Section 2.4 discusses some other related work to high-dimensional data with a focus on feature extraction techniques.

2.1 Feature Selection Methods

Feature selection has been generally viewed as a problem of searching for an optimal subset of features guided by some evaluation measures. For high-dimensional data with hundreds of thousands of features involved, it is usually intractable [Kohavi and John 1997] to find an optimal feature subset and feature selection could be even considered as a **NP**-hard problem [Blum and Rivest 1992]. But various feature selection algorithms have been proposed with the help of informative feature evaluation metrics and refined searching procedures. They have been also proved both efficient in terms of selection process and effective at improving the performance of learning models

built on the selected features.

Typically, the feature selection process can be decomposed into four basic steps, including subset generation, subset evaluation, stopping criterion, and result validation [Dash and Liu 1997]. Subset generation involves a search procedure, which is used for producing candidate feature subsets for further evaluation. Obviously, the search strategy can dominate the time complexity of a feature selection algorithm while the evaluation criterion will guide the choice of features with best desired quality. The process of subset generation and evaluation is repeated until a given stopping criterion is satisfied and then the selected subset is eventually validated on the test cases from synthetic or real world data sets. Feature selection can be found in many data mining areas such as classification, clustering, association analysis and regression analysis. And feature selection is also a well-studied research area in statistics, where it is called in different names, such as variable selection or coefficient shrinkage for regression [Hastie et al. 2001].

2.1.1 Filter, Wrapper and Embedded Models for Feature Selection

Various feature selection algorithms can broadly fall into the filter model, the wrapper model and the embedded model [Kohavi and John 1997], which are classified based on the evaluation criterion used while searching for the desired feature subset. It is noteworthy that the evaluation criteria decides the quality of features with close relation to the performance of learning models built on those selected features.

Filter algorithms use measures of intrinsic data characteristics [Dash et al. 2002; Hall 2000; Liu and Setiono 1996; Yu and Liu 2003] for evaluating the goodness of features. For a given data set D, the algorithm starts the search from a given subset of features S_0(an empty set, a full set, or any randomly selected subset) and searches through the feature space by a particular search strategy, such as sequential

forward search or sequential backward search. Essentially, each generated subset S is evaluated by an independent measure based on intrinsic data characteristics to decide whether it will replace the previous one as the best subset in current iteration. For high-dimensional data, filter algorithms are often preferred due to their computational efficiency. Among the different search strategies, a very simple way of searching is to evaluate each feature independently and form a subset based on the ranking of features. Such univariate methods have been shown effective in some applications [Forman 2003; Li et al. 2004].

Wrapper model is quite similar to the filter model except that it utilizes a different criteria for feature evaluation. To be specifically, wrapper algorithms rely on the performance of a adopted learning algorithm to evaluate the goodness of a subset of features [Caruana and Freitag 1994; Dy and Brodley 2000; Kim et al. 2000; Kohavi and John 1997] instead of an independent measure based on intrinsic data characteristics as the filter model usually does. It searches for features better suited to the learning algorithm aiming to improve learning performance and thus different learning algorithms could lead to different feature selection results. But it also tends to be more computationally expensive than the filter model because of repeatedly building learning models on each candidate subset of features. For example, Kohavi [Kohavi and John 1997] use hill-climbing search engine and best-first search engine to search for an optimal feature subset tailored to two induction algorithms, Naive-Bayes and decision-tree, respectively.

Embedded methods, similar to wrapper methods, also involve a learning algorithm for feature evaluation. However, rather than repeatedly applying the learning algorithm as a black box on every candidate feature subset, embedded methods incorporate a pre-selection step as part of the model training process. Apparently, embedded methods avoid retraining a model from scratch for every possible subset to be

evaluated and they are more efficient than wrapper methods. Examples of embedded methods include decision trees or families of regularized learning algorithms [Hastie et al. 2004; Perkins et al. 2003].

2.1.2 Representative Feature Selection Algorithms

Although the univariate ranking methods are superior in terms of efficiency, they do not work well when features highly correlate or interact with each other since the independent feature evaluation measure itself does not consider feature correlations at all. Various algorithms based on feature subset search evaluate features in a subset together and select a small subset of relevant but non-redundant features [Appice et al. 2004; Ding and Peng 2003; Yu and Liu 2004]. They have shown improved classification accuracy over univariate methods.

Another way of search is to weight all features together according to the concept of maximum margin and form a subset based on top-ranked features [Guyon et al. 2002; Robnik-Sikonja and Kononenko 2003]. An advantage is that optimal weights of features can be estimated by considering features together.

Clustering has recently been applied to feature selection, by clustering features and then selecting one (or a few) representative features from each cluster [Au et al. 2005; Butterworth et al. 2005; Hastie et al. 2001], or simultaneously clustering and selecting features [Jornsten and Yu 2003], to form a final feature subset. Intuitively, clustering features can illuminate relationships among features and facilitate feature redundancy analysis; a feature is more likely to be redundant to some other features in the same cluster than features in other clusters.

Some of the clustering feature selection algorithms [Hastie et al. 2001; Jornsten and Yu 2003] produce feature selection results in the form of a set of feature groups each consisting of features relevant to the class but highly correlated to each other, instead of the traditional form of a single subset of features. Such set of predictive

feature groups not only generalizes well but also provides additional informative group structure for domain experts to further investigate.

2.1.2.1 Univariate Ranking

Univariate ranking is based on the simple ranking scheme of selecting features and it requires the choice of informative evaluation metrics to achieve better performance. Various measures (e.g., Information Gain, Symmetrical Uncertainty, F-statistic, or Pearson correlation) can be used to evaluate and rank the importance or relevance of individual features, and they have shown similar performance in terms of generalization accuracy [Li et al. 2004] and stability under training data variations for high-dimensional data [Kalousis et al. 2007].

2.1.2.2 ReliefF

ReliefF is a simple and efficient feature weighting algorithm which considers all features together in evaluating the relevance of features [Robnik-Sikonja and Kononenko 2003]. Its key idea is to estimate the relevance of features according to how well their values distinguish between samples that are similar to each other. Like univariate ranking, ReliefF assigns similar relevance scores to highly correlated features, and hence do not minimize redundancy in the top-ranked features.

ReliefF is also a representative one among the margin-based feature selection methods. Study on ReliefF can further unveil the relationship between margin-based feature evaluation(adopted by traditional feature selection algorithms) and margin-based instance weighting(proposed in our study).

2.1.2.3 mRMR

Algorithms above produce a ranking which can be used to pick subsets. Many algorithms apply subset search strategies to select a subset of features which are highly correlated to the class but lowly correlated with each other. mRMR (minimum redundancy and maximum relevance) is one of those algorithms which also takes into

account such correlations among features besides relevance. Another reason for its popularity is because of the computational efficiency and flexibility in specifying the number of selected features [Ding and Peng 2003].

mRMR aims to select a subset of features which simultaneously maximize V_F and minimize W_r. In practice, mRMR combines the two criteria into a single criterion function by maximizing the difference or quotient of the two, and applies heuristic sequential forward search to add one feature at a time to the current best subset until a desired number of features is reached.

2.1.2.4 SVM-RFE

SVM-RFE is a popular algorithm to exploit sample margin for evaluating the goodness of features [Guyon et al. 2002]. The main process of SVM-RFE is to recursively eliminate features based on Support Vector Machine(SVM), using the coefficients of the optimal decision boundary to measure the relevance of each feature. At each iteration, it trains a linear SVM classifier, ranks features according to the squared values of feature coefficients assigned by the linear SVM, and eliminates one or more features with the lowest scores. SVM-RFE differentiates the weights of highly correlated features and hence can minimize redundancy in the top-ranked features.

2.1.2.5 HHSVM

Hybrid Hyberized Support Vector Machines(HHSVM) is a recently proposed method which applies to SVM a hybrid of the L1-norm and the L2-norm penalties to select highly correlated features together and has shown more stable than the L1-norm SVM [Wang et al. 2007]. The support vector machine (SVM) is a widely used classification technique, and previous studies have demonstrated its superior classification performance in microarray analysis. However, a major limitation is that SVM can not perform automatic gene selection. HHSVM uses the hinge loss function and the elastic-net penalty. The major benefits include the automatic gene selection and the

grouping effect, where highly correlated genes tend to be selected or removed together.

2.2 Stability of Feature Selection

2.2.1 Stability of Learning Algorithms

Stability, defined as the insensitivity of a method to variations in the training set, has been extensively studied with respect to the learning algorithm itself and was firstly examined by Turney [Turney 1995]. A measure was proposed based on the agreement of classification models produced by the same algorithm when trained on different training sets. The agreement of two classification models is defined as the probability that they will produce the same predictions over all possible instances drawn from a probability distribution $P(X)$. It is noteworthy that instances are drawn from $P(X)$ and not from $P(X, C)$, the joint probability distribution of both class and training instances; the underlying reason is that the agreement of two concepts (classification models) should be examined in all possible inputs. In order to estimate stability, a method of m repetitions of 2-fold cross-validation is used. A classification model is produced from each one of the two folds for each of m repetitions and the two models are then tested on artificial instances drawn by sampling from $P(X)$ and their agreement is computed. Eventually, the final estimation of stability is to get the average over all m runs.

The bias-variance decomposition of the error of classification algorithms [Domingos 2000; Geman et al. 1992] can be considered as another work for defining the stability of learning algorithms. In fact, the variance term quantifies instability of the classification algorithm in terms of classification predictions. Variance measures the percentage of times that the predictions of different classification models, learned from different training sets, for a given instance are different from the typical (average) prediction. Bias-variance decomposition is usually done via bootstrapping, where part of the data is kept as a hold-out test set and the remainder is used to cre-

ate different training sets by using sampling with replacement. The final estimation of variance is also the average over the different bootstrapped samples.

2.2.2 Existing Study on Stable Feature Selection

There exist very limited studies on feature selection stability. Early work on this topic focuses on stability measures and empirical evaluation of the stability of feature selection algorithms [Kalousis et al. 2007; Kuncheva 2007]. Two papers [Davis et al. 2006; Kalousis et al. 2007] studied the stability issue of feature selection under small sample size, and compared a few feature selection algorithms. According to [Kalousis et al. 2007], the stability of a feature selection algorithm is defined as the robustness of the results (e.g., selected subsets, feature weights) the algorithm produces to differences in training sets drawn from the same generating distribution $P(X; C)$. Both papers concluded that algorithms which performed equally well for classification had a wide difference in terms of stability, and suggested empirically choosing the best feature selection algorithm based on both accuracy and stability. It's noteworthy that the feature subset with higher stability score does not necessarily improve the predictive accuracy of learning models built on those selected features under the study in those papers and the motivation of stable feature selection is not well explained.

More recently, two approaches were proposed to improve the stability of feature selection algorithms without sacrificing classification accuracy. Saeys *et al.* studied bagging-based ensemble feature selection [Saeys et al. 2008] which aggregates the results from a conventional feature selection algorithm repeatedly applied on a number of bootstrapped samples of the same training set. Yu *et al.* proposed an alternative approach to tackle the instability problem by exploring the intrinsic correlations among the large number of features. They proposed a group-based stable feature selection framework which identifies groups of correlated features and selects relevant feature groups [Loscalzo et al. 2009; Yu et al. 2008].

2.2.3 Ensemble Feature Selection and Group-based Feature Selection

Since the ensemble feature selection explores the intrinsic relations in the sample space which is comparable with our proposed weighting approach, we include the experiment results for ensemble approach for comparison in our empirical study. Group-based feature selection explores the feature correlations instead which represent another line of study and thus we simply illustrate the idea in this section only.

2.2.3.1 Ensemble Feature Selection

Ensemble method has been both theoretically and empirically proved effective at reducing the instability of learning algorithms while maintaining even improving their prediction performance. By introducing the idea of ensemble feature selection, Saeys *et al.* studied how bagging-based ensemble feature selection [Saeys et al. 2008] affects the stability of feature selection results and further improves the prediction performance of learning models built upon the selected features. More specifically, they aggregates the results from a conventional feature selection algorithm repeatedly applied on a number of bootstrapped samples of the same training set. This approach tackles the instability problem of feature selection by extensively exploring the available training instances; it simulates a training set of larger sample size by creating a number of bootstrapped training sets.

2.2.3.2 Group-based Feature Selection

Feature correlation are usually examined for better selection of non-redundant features in traditional feature selection algorithms, such as mrMR [Ding and Peng 2003]. Yu *et al.* proposed an alternative approach to tackle the instability problem by exploring the intrinsic correlations among the large number of features. They proposed a group-based stable feature selection framework which identifies groups of correlated features and selects relevant feature groups [Loscalzo et al. 2009; Yu et al. 2008]. The

idea of stable feature selection via dense feature groups was first proposed and tested in [Yu et al. 2008]. The idea was motivated by two main observations in the sample space (the feature space defined by a set of samples in a training set). Firstly, the dense core regions (peak regions), measured by probabilistic density estimation, are stable with respect to sampling of the dimensions (samples). Another observation is that the features near the core region are highly correlated to each other, and thus should have similar relevance scores w.r.t. some class labels, assuming the class labels are locally consistent.

2.3 Margin Based Feature Selection

2.3.1 Sample Margin vs. Hypothesis Margin

Another line of research closely related to our work is margin based feature selection. Essentially, the goodness of features are evaluated through margins. In fact, margins [Cortes and Vapnik 1995] measure the confidence of a classifier w.r.t. its decision, and have been used both for theoretical generalization bounds and as guidelines for algorithm design.

There are two natural ways of defining the margin of a sample w.r.t. a hypothesis [Crammer et al. 2002]. The more common type, *Sample-margin* (SM) measures the distance between the instance and the decision boundary induced by the classifier. Support Vector Machines or SVMs [Cortes and Vapnik 1995] find the separating hyperplane with the largest sample-margin. In Barlett's paper [Bartlett and J. 1999], he also discusses the distance between instances and the decision boundary and uses the sample-margin to derive generalization bounds.

The hypothesis-margin (HM) is an alternative definition of margin. It requires the existence of a distance measure on the hypothesis class. More specifically, the margin of an hypothesis with respect to an instance is the distance between the hypothesis and the closest hypothesis that assigns alternative label to the given instance [Gilad-

Bachrach et al. 2004]. For example AdaBoost [Freund and Schapire 1997] uses this type of margin with the L1-norm as the distance measure among hypotheses.

Figure 2.1 illustrates the difference between Sample Margin and Hypothesis Margin. To make it simple, Sample Margin measures how much an instance can travel before it hits the decision boundary while Hypothesis margin measures how much the hypothesis can travel before it hits an instance.

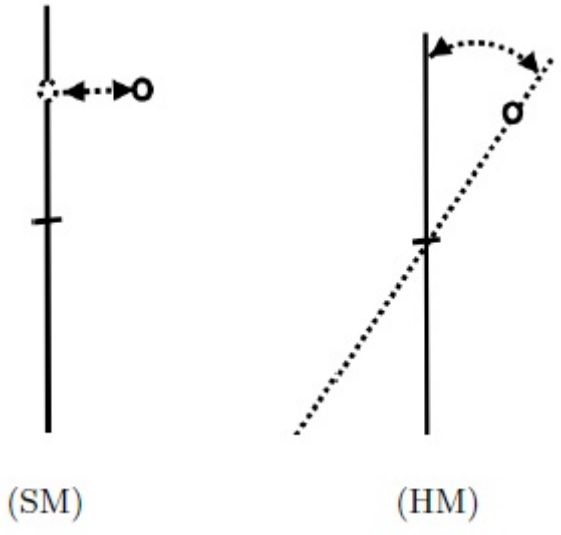

Figure 2.1. A simple illustration of Sample Margin (SM) and Hypothesis Margin (HM).

2.3.2 Feature Evaluation via Exploiting Margin

Various feature selection algorithms have been developed under the large margin (SM or HM) principles. These algorithms evaluate the importance of features according to their respective contributions to the margins, and have exhibited both nice theoretical properties and good generalization performance. However, the stability of these algorithms is an issue for training data with small sample size but high dimensionality.

Sample margin is an important property when making the final decision on the optimal hyperplane of high-dimensional space and thus the feature importance achieved by the choice of hyperplane will exert immediate influence on the predictive accu-

racy. This is essentially why SVM-based feature selection have demonstrated very good results on improving the classification performance [Bradley and Mangasarian 1998; Guyon et al. 2002; Wang et al. 2007]. Hypothesis margin is determined by the hypothesis chosen in the algorithm and k nearest neighbor(KNN) is the mostly studied. Relief family of algorithms [Gilad-Bachrach et al. 2004; Robnik-Sikonja and Kononenko 2003; Sun and Li 2006] utilize the hypothesis margin based on KNN and have been well studied. We have also introduced SVM-RFE and ReliefF as representative feature selection algorithms in Section 2.1.2.

Our study also employs the concept of margins in the proposed margin based instance weighting algorithm. In contrast with margin based feature selection algorithms (e.g., ReliefF [Robnik-Sikonja and Kononenko 2003]) which directly use margins to weight *features*, our algorithm exploits the discrepancies among the margins at various instances to weight *instances*. Our algorithm acts as a preprocessing step to produce a weighted training set which can be input to any feature selection algorithm capable of handling weighted instances.

2.4 Other Related Work on High-Dimensional Data

In machine learning and statistics, feature selection is also called variable selection or attribute selection. It is just one type of dimension reduction techniques (the process of reducing the number of random variables under consideration). Another widely used dimensionality reduction technique is feature extraction which transforms the input data into a reduced representation set of features. Similar to feature selection, feature extraction aims at reducing the dimensionality of the feature space to improve the prediction performance while reducing model complexity. But it differentiates itself from feature selection by constructing combinations of original variables and generating new or latent variables while feature selection retains a subset of features from the original space. The process of data transformation may be linear, as in prin-

cipal component analysis (PCA). Moreover, PCA can also be employed in a nonlinear way and there also exist many other non-linear feature extraction techniques.

2.4.1 Linear Feature Extraction Techniques

Principal component analysis (PCA) is one of the mostly used linear techniques for dimensionality reduction. The purpose of this approach is to derive a new set of variables in a decreasing order of importance which are uncorrelated to each other and also linear combinations of the original variables. A linear mapping of the data to a lower dimensional space is performed by PCA in such a way that the variance of the data in the low-dimensional representation is maximized.

Practically, the approach requires the construction of the correlation matrix on the data from which the eigenvectors are computed [Fukunaga 1990]. The first few eigenvectors can often be interpreted in terms of the large-scale physical behavior of the system. Essentially, the original space with high dimensionality has been reduced to the space spanned by a few eigenvectors. Although there are data loss during this transformation, the most important variance among the feature space is retained to some extent. PCA does not make assumptions about the existence or otherwise of groupings within the data and is accordingly considered as an unsupervised feature extraction technique.

PCA is closely related to factor analysis [Child 2006]. Factor analysis typically incorporates more domain specific assumptions about the underlying structure and solves eigenvectors of a slightly different matrix.

2.4.2 Non-linear Feature Extraction Techniques

Although there are a lot of modern methods for nonlinear dimensionality reduction, most of them originated theoretically and algorithmically from PCA or K-means. Principal component analysis can be extended in a nonlinear way by introducing the kernel trick [Aizerman et al. 1964]. The idea behind it is to construct nonlinear

mappings which can maximize the variance in the data.

Besides nonlinear PCA, some other popular nonlinear feature extraction techniques include manifold learning such as locally linear embedding (LLE) [Roweis and Saul 2000], Laplacian eigenmaps [Belkin 2003] and Hessian LLE [Donoho and Grimes 2003]. Essentially, a cost function which can retain local properties of the data is used for constructing a low-dimensional data representation and they can also be viewed as defining a graph-based kernel for Kernel PCA. Semidefinite programming [Weinberger and Saul 2004] tried to learn the kernel instead of defining a fixed kernel and the prominent example of this technique is to use maximum variance unfolding (MVU), which preserve all pairwise distances between nearest neighbors (in the inner product space), while maximizing the distances between points that are not nearest neighbors.

The minimization of a cost function which measures differences between distances in the input and output spaces can also achieve neighborhood preservation, such as classical multidimensional scaling (which is identical to PCA), Isomap (which uses geodesic distances in the data space), diffusion maps (which uses diffusion distances in the data space), t-SNE (which minimizes the divergence between distributions over pairs of points), and curvilinear component analysis [Maaten et al. 2009].

3 Theoretical Framework for Stable Feature Selection

In Section 3.1, we formally define the stability of feature selection from a sample variance perspective, present a bias-variance decomposition of feature selection error, and discuss the relationship between the stability and the accuracy of feature selection based on this decomposition. In Section 3.2, we further show that for feature selection algorithms which can be viewed as Monte Carlo estimators, their stability depends on the sample size and can be improved by variance reduction techniques such as importance sampling.

3.1 Bias-Variance Decomposition of Feature Selection Error

3.1.1 Bias, Variance, Error of Feature Selection

Let $D = \{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ be a training set of n labeled instances, where $\mathbf{x} \in \Re^d$, defined by d features $X_1, ..., X_d$, and y is the value of the class variable Y. In general, the result of a feature selection algorithm $\mathcal{F}$ on a training set D can be viewed as a vector $\mathbf{r} = (r_1, ..., r_d)$, where r_j $(1 \leq j \leq d)$ is the *estimated* relevance score of feature X_j assigned by $\mathcal{F}$. Let $\mathbf{r}^* = (r_1^*, ..., r_d^*)$ be a vector indicating the *true* relevance score of each feature to the class. In this dissertation, we focus our discussion on feature weighting algorithms, and adopt the commonly used squared loss function $(r_j^* - r_j)^2$ to measure the error made by $\mathcal{F}$ on feature X_j. When there is no risk of ambiguity, we will drop the subscript $_j$ and use r^* or r to represent the true or estimated relevance

score of any feature X, respectively.

For the same feature X, a feature selection algorithm $\mathcal{F}$ in general produces different estimated relevance scores r based on different training sets D. Therefore, we speak of D as a random variable and use $r(D)$ to represent the estimated relevance score of feature X based on a given training set. In turn, the resulting error $(r^* - r(D))^2$ is also a function of D. To evaluate the overall performance of $\mathcal{F}$, the quantity of interest is the *expected loss (error)*, $EL(X)$, defined as:

$$EL(X) = \mathbb{E}[(r^* - r(D))^2] = \sum_{D \in \mathscr{D}} (r^* - r(D))^2 p(D) \ , \tag{3.1}$$

where $\mathscr{D}$ is the set of all possible training sets of size n drawn from the same underlying data distribution, and $p(D)$ is the probability mass function on $\mathscr{D}$.

Let $\mathbb{E}(r(D)) = \sum_{D \in \mathscr{D}} r(D)p(D)$ be the expected value of the estimates for feature X over $\mathscr{D}$. The *bias* of a feature selection algorithm $\mathcal{F}$ on a feature X is defined as:

$$Bias(X) = [r^* - \mathbb{E}(r(D))]^2 \ . \tag{3.2}$$

The *variance* of a feature selection algorithm $\mathcal{F}$ on a feature X is defined as:

$$Var(X) = \mathbb{E}[r(D) - \mathbb{E}(r(D))]^2 = \sum_{D \in \mathscr{D}} [r(D) - \mathbb{E}(r(D))]^2 p(D) \ . \tag{3.3}$$

Apparently, these formal definitions of error, bias and variance of a given feature selection algorithm above provide a more in-depth view of the different properties of feature selection itself and shed lights on their intrinsic relations for evaluating the goodness of feature selection algorithms from the theoretical perspective.

3.1.2 A Formal Decomposition of Feature Selection Error

Following the above definitions on the expected loss, bias, and variance, for any feature X, we have the following standard decomposition of the expected loss:

$$EL(X) = Bias(X) + Var(X) \ .$$

Intuitively, the bias reflects the loss incurred by the central tendency of $\mathcal{F}$, while the variance reflects the loss incurred by the fluctuations around the central tendency in

response to different training sets.

Extending the above definitions to the entire set of features, we can speak of the average loss, average bias, and average variance, and have the following decomposition among the three:

$$\frac{1}{d}\sum_{j=1}^{d} EL(X_j) = \frac{1}{d}\sum_{j=1}^{d} Bias(X_j) + \frac{1}{d}\sum_{j=1}^{d} Var(X_j) \ . \tag{3.4}$$

The average variance component naturally quantifies the sensitivity or *instability* of a feature selection algorithm under training data variations; lower average variance value means higher stability of the algorithm. We will use the average variance as one of the stability measures in the empirical study. The above bias-variance decomposition is based on feature weighting algorithms and squared loss function, and can be extended to feature subset selection algorithms under zero-one loss function in future study.

3.1.3 Relationship between Stability of Feature Selection and Prediction Accuracy

The above bias-variance decomposition reveals the relationship between the accuracy (the opposite of loss) and stability (the opposite of variance). Reducing either the bias or the variance alone does not necessarily reduce the error because the other component may increase, but a better tradeoff between the bias and the variance does. One thing to note at this point is that the loss of feature selection in the above decomposition is measured with respect to the true relevance of features, not the generalization error of the model learned based on the selected features. The former, in theory, is consistent with the latter; a perfect weighting of the features leads to an optimal feature set and hence an optimal Bayesian classifier [Koller and Sahami 1996]. However, in practice, the generalization error depends on both the loss of feature selection and the bias-variance properties of the learning algorithm itself. An in-depth study of how the stability of feature selection affects the bias-variance

properties of various learning algorithms would be interesting, but is out of the scope of this dissertation.

Nevertheless, the framework presented here sheds lights on the relationship of feature selection and the predictive accuracy based on the selected features. Existing studies on stable feature selection [Kalousis et al. 2007; Saeys et al. 2008] showed that different feature selection algorithms performed differently w.r.t. stability and predictive accuracy, and there was no clear winner in terms of both measures. They suggested a tradeoff between the stability and predictive accuracy. To pick the best algorithm for a given data set, a user could use a joint measure which weights the two criteria based on the user's preference on higher accuracy or higher stability. In contrast to the previous studies, our theoretical framework suggests that one does not have to sacrifice predictive accuracy in order to get more stable feature selection results. A better tradeoff between the bias and variance of feature selection can lead to more stable feature selection results, while maintaining or even improving predictive accuracy based on selected features. In the next section, we show a general principle to achieve such a better tradeoff for feature weighting algorithms.

3.2 Variance Reduction via Importance Sampling

3.2.1 Variance, Bias and Error of Monte Carlo Estimator

Consider a (possibly multidimensional) random variable X having probability density function $f(x)$ on a set of values $\mathscr{X}$. Then the expected value of a function g of X is

$$\mathbb{E}(g(X)) = \int_{x \in \mathscr{X}} g(x) f(x) dx \ .$$

The variance of $g(X)$ is

$$Var(g(X)) = \int_{x \in \mathscr{X}} [g(x) - \mathbb{E}(g(X))]^2 f(x) dx \ .$$

If we were to randomly take an n-sample of X's and compute the mean of $g(x)$ over the sample, then we would have the Monte Carlo *estimate*

$$\widetilde{g_n}(x) = \frac{1}{n}\sum_{i=1}^{n} g(x_i)$$

of $\mathbb{E}(g(X))$. We could, alternatively, speak of the random variable

$$\widetilde{g_n}(X) = \frac{1}{n}\sum_{i=1}^{n} g(X_i)$$

as the Monte Carlo *estimator* of $\mathbb{E}(g(X))$. Note that $\widetilde{g_n}(X)$ is unbiased for $\mathbb{E}(g(X))$:

$$\mathbb{E}(\widetilde{g_n}(X)) = \mathbb{E}(g(X)) \ .$$

The variance of $\widetilde{g_n}(X)$ is:

$$Var(\widetilde{g_n}(X)) = \frac{Var(g(X))}{n} \ .$$

Many feature weighting algorithms such as the Relief family of algorithms and SVM-based algorithms decide the relevance score for each feature based on aggregating the scores over all instances in a training set. We can view such algorithm as a Monte Carlo estimator in the above context. Let $g(X)$ be the weighting function associated with a feature weighting algorithm $\mathcal{F}$, which assigns a relevance score to feature X based on each sampled value of X. $\mathbb{E}(g(X))$ represents the relevance score of X assigned by $\mathcal{F}$ based on the entire distribution. $\widetilde{g_n}(x)$, the Monte Carlo estimate of $\mathbb{E}(g(X))$, represents the estimated relevance score of X assigned by $\mathcal{F}$ based on a training set D of size n. To measure the expected loss, bias, and variance of $\mathcal{F}$, Eqs. (1), (2), and (3) defined in the previous section can be respectively rewritten as follows:

$$EL(X) = \mathbb{E}[(r^* - \widetilde{g_n}(X))^2] \ , \tag{3.5}$$

$$Bias(X) = [r^* - \mathbb{E}(\widetilde{g_n}(X))]^2 = [r^* - \mathbb{E}(g(X))]^2 \ , \tag{3.6}$$

$$Var(X) = \mathbb{E}[\widetilde{g_n}(X) - \mathbb{E}(\widetilde{g_n}(X))]^2 = \frac{Var(g(X))}{n} \ . \tag{3.7}$$

The above formulations show that given a data distribution, $Bias(X)$ depends on the feature selection algorithm, while $Var(X)$ depends on both the feature selection algorithm and the sample size of the training set. Different feature selection algorithms may have different bias and variance properties, and hence different errors. For the same feature selection algorithm, increasing the sample size leads to a reduction of both variance and the expected error of feature selection. In reality, increasing the sample size could be impractical or very costly in many applications. For example, in microarray data analysis, each microarray sample is from the tissue of a patient, which is usually hard to obtain and costly to perform experiments on. Fortunately, there are numerous ways to reduce the variance of a Monte Carlo estimator without increasing the sample size.

3.2.2 How Importance Sampling Works for Variance Reduction

Importance sampling is one of the commonly used variance reduction techniques [Rubinstein 1981]. Intuitively, importance sampling is about choosing a good distribution from which to simulate one's random variables. It can be formally stated by Theorem 3.1 below (See [Rubinstein 1981] p. 123 for the proof).

Theorem 3.1. *Let $h(x)$ be the probability density function for the random variable X which takes values only in $\mathscr{A}$ so that $\int_{x\in\mathscr{A}} h(x)dx = 1$. Then*

$$\int_{x\in\mathscr{A}} g(x)dx = \int_{x\in\mathscr{A}} \frac{g(x)}{h(x)}h(x)dx = \mathbb{E}_h\left(\frac{g(X)}{h(X)}\right) \ , \tag{3.8}$$

so long as $h(x) \neq 0$ for any $x \in \mathscr{A}$ for which $g(x) \neq 0$. This gives a Monte Carlo estimator:

$$\widetilde{g_n^h}(X) = \frac{1}{n}\sum_{i=1}^{n}\frac{g(X_i)}{h(X_i)} \quad where \quad X_i \sim h(x).$$

$Var(\widetilde{g_n^h}(X))$ is minimized when $h(x) \propto |g(x)|$.

A good importance sampling function $h(x)$ will be one that is as close as possible

to being proportional to $|g(x)|$. Since $g(x)$ in Eq. 3.8 can be any arbitrary function, we have

$$\mathbb{E}(g(X)) = \int_{x \in \mathscr{X}} g(x)f(x)dx = \mathbb{E}_h\left(\frac{g(x)f(x)}{h(x)}\right) .$$

To reduce $Var(\widetilde{g_n^h}(X))$, one can go searching a suitable $h(x)$ that is close to proportional to $|g(x)f(x)|$. This is a rather difficult task since $g(x)f(x)$ is unknown and has to be estimated from an empirical distribution.

3.2.3 Instance Weighting: An Empirical Alternative to Importance Sampling

In practice, it is rather difficult to find such a function $h(x)$, especially in high-dimensional spaces. Nevertheless, the theory of importance sampling suggests that in order to reduce the variance of a Monte Carlo estimator, instead of performing i.i.d. sampling, we should increase the number of instances taken from regions which contribute more to the quantity of interest and decrease the number of instances taken from other regions. When given only the empirical distribution in a training set, although we cannot redo the sampling process, we can simulate the effect of importance sampling by increasing the weights of instances taken from more important regions and decreasing the weights of those from other regions. Therefore, the problem of variance reduction for feature selection boils down to finding an empirical solution of instance weighting.

Apparently, a good instance weighting scheme should discriminate the instances w.r.t their respective influence on the stability of feature selection results since we optimize for a solution to reduce the variance of feature selection. Motivated by this intuition, we propose the margin based instance weighting approach in our empirical framework of stable feature selection described in next chapter with details.

4 Empirical Framework: Margin Based Instance Weighting

The empirical framework is motivated by importance sampling introduced in Chapter 3, one of the commonly used variance reduction techniques [Rubinstein 1981]. In this chapter, we formally propose an empirical framework of instance weighting for variance reduction. As shown in Figure 4.1, the proposed framework differs from conventional feature selection which directly works on a given training set and consists of three key components:

- transforming the original feature space into margin vector feature space which enables the estimation of the importance of instances w.r.t. feature evaluation;
- weighting each training instance according to its importance in the margin vector feature space;
- applying a given feature selection algorithm on the original feature space with weighted instances.

Each of these three key components will be elaborated in turn. Section 4.1 introduces the main ideas and technical details of margin vector feature space. Section 4.2 explains the instance weighting scheme under the transformed space. Section 4.3 combines the first two components into an efficient algorithm of margin based instance weighting. Section 4.4 further explores an iterative approach for margin based instance weighting. In the third component, how to exploit weighted instances in fea-

ture selection depends on the specific choice of a feature selection algorithm, and will be discussed in next chapter where two representative feature selection algorithms are extended to take into account instance weights.

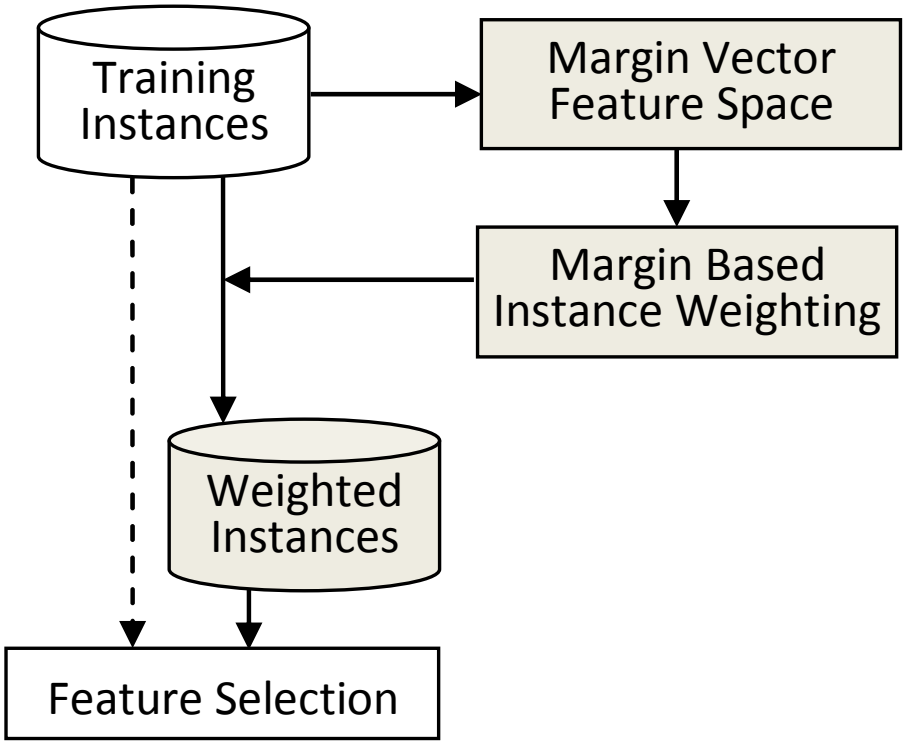

Figure 4.1. An empirical framework of margin based instance weighting for stable feature selection, consisting of three key components connected by solid arrows.

4.1 Margin Vector Feature Space

4.1.1 Introduction of Margin Vector Feature Space

Margins measure the confidence of a classifier w.r.t. its decisions, and have been used both for theoretical generalization bounds and as guidelines for algorithm design [Cortes and Vapnik 1995]. There are two natural ways of defining the margin of an instance w.r.t. a classifier [Crammer et al. 2002]. *Sample margin* measures the distance between an instance and the decision boundary of a classifier. Support Vector Machine (SVM) [Cortes and Vapnik 1995], for example, uses this type of margin; it finds the separating hyper-plane with the largest sample margin for support

vectors. An alternative definition, *hypothesis margin*, measures the distance between the hypothesis of an instance and the closest hypothesis that assigns alternative label to the instance. Hypothesis margin requires a distance measure between hypotheses. For example, AdaBoost [Freund and Schapire 1997] uses this type of margin with the L_1-norm as the distance measure between hypotheses. Feature selection algorithms developed under the large margin principles [Guyon et al. 2002; Gilad-Bachrach et al. 2004] evaluate the relevance of features according to their respective contributions to the margins.

For 1-Nearest Neighbor (1NN) classifier, [Crammer et al. 2002] proved that (i) the hypothesis margin lower bounds the sample margin; and (ii) the hypothesis margin of an instance $\mathbf{x}$ w.r.t. a training set D can be computed by the following formula under L_1-norm:

$$\theta_D(\mathbf{x}) = \frac{1}{2}\left(||\mathbf{x} - \mathbf{x}^M|| - ||\mathbf{x} - \mathbf{x}^H||\right) \ ,$$

where $\mathbf{x}^H$ and $\mathbf{x}^M$ represent the nearest instances (called Hit and Miss) to $\mathbf{x}$ in D with the same and opposite class labels, respectively.

Since hypothesis margin is easy to compute and large hypothesis margin ensures large sample margin, we focus on hypothesis margin in this dissertation. We will speak of hypothesis margin simply as margin in the rest of this dissertation. In our framework of instance weighting, we employ the concept of margin in a different way. By decomposing the margin of an instance along each dimension, the instance in the original feature space can be represented by a new vector (called *margin vector*) in the *margin vector feature space* defined as follows.

Definition 4.1. Let $\mathbf{x} = (x_1, ..., x_d)$ be an instance from training set (original feature space) D drawn from real space $\Re^d$, and $\mathbf{x}^H$ and $\mathbf{x}^M$ represent the nearest instances to $\mathbf{x}$ with the same and opposite class labels, respectively. For each $\mathbf{x} \in \Re^d$, $\mathbf{x}$ can be

mapped to $\mathbf{x}'$ in a new feature space D' according to:

$$x'_j = |x_j - x_j^M| - |x_j - x_j^H| \ , \tag{4.1}$$

where x'_j is the jth coordinate of $\mathbf{x}'$ in the new feature space D', and x_j, x_j^M, or x_j^H is the jth coordinate of $\mathbf{x}$, $\mathbf{x}^H$ or $\mathbf{x}^M$ in $\Re^d$, respectively. Vector $\mathbf{x}'$ is called the margin vector of $\mathbf{x}$, and D' is called the margin vector feature space.

In essence, $\mathbf{x}'$ captures the local profile of feature relevance for all features at $\mathbf{x}$. The larger the value of x'_j, the more feature X_j contributes to the margin of instance $\mathbf{x}$. Thus, the margin vector feature space captures local feature relevance profiles (margin vectors) for all instances in the original feature space.

4.1.2 Insight from An Illustrative Example

Figure 4.2 illustrates the concept of margin vector feature space through a 2-d example. Each instance in the original feature space is projected into the margin vector feature space according to Eq. (4.1). We can clearly see that the three instances highlighted by circles exhibit largely different outlying degrees in the two feature spaces. Specifically, the two instances in the solid circles are close to the center of all instances in the original space, but are far apart from the center in the transformed space. The one in the dashed circle shows the opposite trend; it appears as an outlier in the original feature space, but becomes close to the center in the transformed space. Overall, the margin vector feature space captures the distance among instances w.r.t. their margin vectors (instead of feature values in the original space), and enables the detection of instances that largely deviate from others in this respect.

To weight the importance of features X_1 and X_2 in the example, one intuitive idea is to aggregate all margin vectors along each dimension, as adopted by the well-known Relief algorithm [Robnik-Sikonja and Kononenko 2003]. Since the two instances in the solid circles exhibit distinct margin vectors from the rest of the

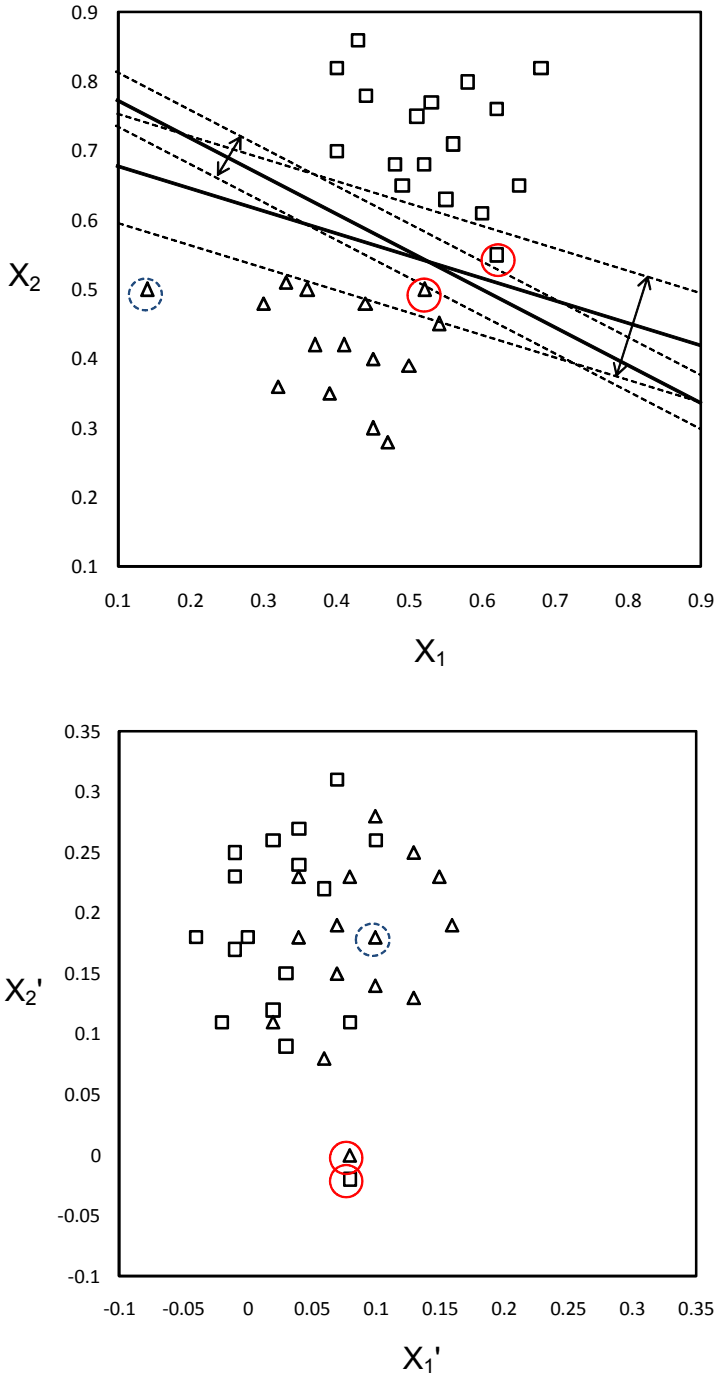

Figure 4.2. An illustrative example for Margin Vector Feature Space. Each data point in the original feature space (left) is projected to the margin vector feature space (right) according to a decomposition of its hypothesis margin along each dimension in the original feature space.

instances, the presence or absence of these instances will contribute to large variance of the aggregate feature weights under training data variations. Another approach, as used by SVM-RFE [Guyon et al. 2002], is to weight features based on the magnitude of feature coefficients in the optimal decision boundary of a linear SVM classifier. As shown in the example, the decision boundaries with or without the two instances in the solid circles show different slopes, resulting in different weights of features X_1 and X_2. Therefore, such instances also cause variance in feature weighting by SVM. By identifying and reducing the emphasis on these outlying instances, more stable results can be produced from a feature selection algorithm. Although those instances with higher outlying degree in the margin vector feature space contribute more to the instability of feature selection results, they may have positive effects on the feature selection accuracy and should not be considered as outliers from the that perspective. In our study, we focus on the improvement of feature selection stability and how improved stability affects the predictive accuracy of learning models. Therefore, we would consider those unstable instances as outliers in our approach under study. But it is noteworthy that real effects of those instances can only be validated through real-world applications. In the next section, we will discuss how to exploit the margin vector feature space to weight instances in order to alleviate the affect of training data variations on feature selection results. In Chapter 5, we will further discuss how to incorporate weighted instances by ReliefF and SVM-RFE.

4.1.3 Extension for Hypothesis Margin of 1NN

The above definition and example of margin vector feature space only consider one nearest neighbor from each class. To reduce the affect of noise or outliers in the training set on the transformed feature space, multiple nearest neighbors from each class can be used to compute the margin vector of an instance. In this work, we consider all neighbors from each class for a given instance. Eq. (4.1) can then be

extended to:

$$x'_j = \sum_{l=1}^{m} |x_j - x_j^{M_l}| - \sum_{l=1}^{h} |x_j - x_j^{H_l}| \ , \tag{4.2}$$

where $x_j^{H_l}$ or $x_j^{M_l}$ denotes the jth component of the lth neighbor to $\mathbf{x}$ with the same or different class labels, respectively. m or h represents the total number of misses or hits ($m + h$ equals the total number of instances in the training set excluding the given instance).

A further extension of the above formula is to differentiate those nearest neighbor from each class by assigning them weights according to their individual distance from the instance respectively. It can be extended as below:

$$x'_j = \sum_{l=1}^{m} w^{M_l} |x_j - x_j^{M_l}| - \sum_{l=1}^{h} w^{H_l} |x_j - x_j^{H_l}| \ , \tag{4.3}$$

where w^{M_l} denotes the weight of each miss and w^{H_l} denotes the weight of each hit respectively.

Apparently, greater distance indicates less influence on assigning the instance to the corresponding hypothesis dominated by that miss or hit respectively. Therefore, we opt for weighting the miss or hit with greater distance from the instance lower than those with less distance from the instance. To quantify the weight score in terms of distance value, we adopt the use of the inverse of absolute distance value to reflect this intuition. w^{M_l} is defined as:

$$w^{M_l} = \frac{\frac{1}{dist(\mathbf{x},\mathbf{x}^{M_l})}}{\sum_{i=1}^{m} \frac{1}{dist(\mathbf{x},\mathbf{x}^{M_i})}} \ , \tag{4.4}$$

while w^{H_l} is defined as:

$$w^{H_l} = \frac{\frac{1}{dist(\mathbf{x},\mathbf{x}^{H_l})}}{\sum_{i=1}^{h} \frac{1}{dist(\mathbf{x},\mathbf{x}^{H_i})}} \ , \tag{4.5}$$

From the above definitions, we can clearly see that the weight is also normalized by the sum of inverse of absolute distance values for all nearest misses and all nearest hits respectively.

We experimented with both of the non-weighted and weighted hypothesis margin of KNN defined above and realized there is trivial difference in terms of their influence on the final margin vector feature space. Therefore we adopted and reported for the non-weighted extension in our study for efficiency.

4.2 Margin Based Instance Weighting

Recall the intuitions of importance sampling for variance reduction discussed in Section 3.2. In order to reduce the variance of a Monte Carlo estimator, we should increase the number of instances taken from regions which contribute more to the quantity of interest and decrease the number of instances taken from other regions. When given only an empirical distribution, importance sampling can be simulated by instance weighting. In the context of feature selection, in order to reduce the variance in feature weighting, we should increase the weights of instances from regions which contribute more to the aggregate feature weights and decrease the weights of instances from other less important (or outlying) regions.

The margin vector feature space introduced above enables instance weighting w.r.t. the contribution of each instance in feature weighting. As illustrated in Figure 4.2, in the margin vector feature space, the central mass region containing most of the instances is clearly more important in deciding the aggregate feature weight for X_1 and X_2 than the outlying region with a couple of outliers. Following the ideas of importance sampling, instances with higher outlying degrees should be assigned lower instance weights in order to reduce the variance of feature weighting under training data variations. To quantitatively evaluate the outlying degree of each instance $\mathbf{x}$ based on its margin vector $\mathbf{x}'$, we measure the average distance of $\mathbf{x}'$ to all other margin vectors. Obviously, instances in sparse regions will have greater average distance, while instances in dense regions will have shorter average distance. Other alternative weighting schemes can be explored, although this simple heuristic works well accord-

Algorithm 1 Margin Based Instance Weighting

Input: training data $D = \{\mathbf{x}_i\}_{i=1}^n$
Output: weight vector $\mathbf{w} = (w_1, ..., w_n)$ for all instances in D
// Margin Vector Feature Space Transformation
for $i = 1$ **to** n **do**
  for $j = 1$ **to** d **do**
    For $\mathbf{x}_i$, compute $x'_{i,j}$ according to Eq. (4.2)
  end for
end for
// Margin Based Instance Weighting
Calculate and store pair-wise distances among all margin vectors $\mathbf{x}'_i$
for $i = 1$ **to** n **do**
  For $\mathbf{x}_i$, compute its weight $w(\mathbf{x}_i)$ according to Eq. (4.6)
end for

ing to our empirical study. Specifically, the weight for an instance $\mathbf{x}$ is determined by the normalized inverse average distance of its margin vector to all other margin vectors, as in the following formula:

$$w(\mathbf{x}) = \frac{1/\overline{dist}(\mathbf{x}')}{\sum_{i=1}^{n} 1/\overline{dist}(\mathbf{x}'_i)} \ , \tag{4.6}$$

where

$$\overline{dist}(\mathbf{x}') = \frac{1}{n-1} \sum_{p=1, \mathbf{x}'_p \neq \mathbf{x}'}^{n-1} dist(\mathbf{x}', \mathbf{x}'_p) \ .$$

4.3 Explanation and Discussion on the Algorithm

Algorithm 1 combines the processes of margin vector feature space transformation and margin based instance weighting into a single algorithm. Given a training data set, the algorithm first projects all instances in the original feature space to their margin vectors in the margin vector feature space according to Eq. (4.2). It then weights instances according to Eq. (4.6) based on the transformed feature space. The algorithm outputs instance weights for all instances, which can be incorporated by a feature selection algorithm.

Both feature space transformation and instance weighting involve distance computation along all features for all pairs of instances: the former in the original feature

space, and the latter in the margin vector feature space. Since these computations dominate the time complexity of the algorithm, the overall time complexity of the algorithm is $O(n^2 * d)$, where n is the sample size and d is the number of features in a training set. Therefore, the algorithm is very efficient for high-dimensional data with small sample size (i.e., $n << d$).

4.4 Iterative Margin Based Instance Weighting

Recall the weighted Hypothesis Margin of KNN formulated in 4.1.3 and we can see that each nearest miss or nearest hit is assigned a normalized weight based on their respective distance from the instance. It's also noteworthy that our margin based instance weighting framework is essentially used for producing the weight score for each instance and intuitively the weights produced after one run of our approach on the training data can be used as the weights in the formula of weighted hypothesis margin of KNN. Apparently, a different set of weights will be produced afterwards and iteratively reused until it converges to a set of weights without change any more. We rewrite the formula 4.7 as below:

$$x'_j = \sum_{l=1}^{m} W(M_l)|x_j - x_j^{M_l}| - \sum_{l=1}^{h} W(H_l)|x_j - x_j^{H_l}| \ , \tag{4.7}$$

where $W(*)$ indicates the weight score assigned to each miss M_l or each hit H_l by the previous iteration of margin based instance weighting algorithm, respectively.

Intuitively, the margin vector feature space generated at current iteration would be different after taking into account the instance weights produced by previous iteration and the difference will eventually be reflected on a set of new instance weights after applying the algorithm once more. This is an iterative procedure and it will stop until the weights converge at some point.

We also experimented with this iterative approach and observed that there exists little difference between the instance weights produced by the one-run approach and

the final instance weights produced by the iterative approach after a quick convergence. Therefore, we adopt the one-run instance weighting algorithm in our empirical study for efficiency.

Our findings also suggest that the instance weighting approach based on margin vector feature space is insensitive to small variations to the instance weights in the original feature space, which indicates that the approach can improve the stability of feature selection without introducing any other instability issue resulted from the approach itself. It can definitely further strengthen our confidence on the efficiency and effectiveness of the proposed approach.

5 General Experimental Setup

This chapter serves as an introduction to our experimental study on the proposed theoretical and empirical frameworks on stable feature selection. Section 5.1 describes the objectives of our empirical study and outlines the contents in following chapters. Section 5.2 provides details on the methods in comparison, including SVM-RFE and ReliefF and also their ensemble version. Moreover, it explains in details how to integrate the proposed instance weighting approach into those algorithms. Section 5.3 introduces both the subset stability measures and classification performance measures used in our experiments.

5.1 Outline of Empirical Study

The objective of our empirical study is threefold:

- to demonstrate the bias-variance decomposition proposed in Chapter 3;
- to verify the effectiveness of the proposed instance weighting framework on variance reduction;
- to study the impacts of variance reduction on the stability and predictive performance of the selected subsets.

The remaining sections in this chapter describe the methods in comparison and evaluation measures used in our experiments. In Chapter 6, using synthetic data with prior knowledge of the true relevance of features, we demonstrate the bias-variance

decomposition based on the widely adopted SVM-RFE algorithm. We further show that the proposed instance weighting framework significantly reduces the variance of feature weights assigned by SVM-RFE, and consequently, improves both the stability and the predictive accuracy of the selected feature subsets. Moreover, we study the effects of sample size on feature selection and the proposed instance weighting framework. In Chapter 7, using real-world microarray data sets, we verify the effectiveness of the instance weighting framework on variance reduction and stability improvement for both SVM-RFE and ReliefF algorithms. Furthermore, we show that the instance weighting framework is more effective and efficient than the ensemble feature selection framework.

5.2 Methods in Comparison

We choose SVM-RFE and ReliefF as the baseline algorithms for experimental study. We evaluate the effectiveness of the proposed instance weighting framework for these two algorithms. Furthermore, we compare the instance weighting framework with the ensemble framework using SVM-RFE and ReliefF as the base algorithms.

5.2.1 Baseline Algorithms: SVM-RFE and ReliefF

SVM-RFE [Guyon et al. 2002] is chosen as a baseline algorithm because of its wide adoption in high-dimensional data analysis. The main process of SVM-RFE is to recursively eliminate features of lowest ranks, using SVM to rank features. Starting from the full set of features, at each iteration, the algorithm trains a linear SVM classifier based on the remaining set of features, ranks features according to the magnitude of feature coefficients in the optimal hyperplane, and eliminates one or more features with the lowest ranks. This recursive feature elimination (RFE) process stops until all features have been removed or a desired number of features is reached. Our implementation of SVM-RFE is based on Weka's [Witten and Frank 2005] implementation

of soft-margin SVM using linear kernel and default C parameter. As suggested by its debut work, 10 percent of the remaining features are eliminated at each iteration to speed up the RFE process.

ReliefF [Robnik-Sikonja and Kononenko 2003] is chosen as another representative algorithm for margin based feature selection. It is a simple and efficient feature weighting algorithm which considers all features together in evaluating the relevance of features. The main idea of ReliefF is to weight features according to how well their values distinguish between instances that are similar to each other. Specifically, for a two-class problem, the weight for each feature X_j is determined as follows:

$$W(X_j) = \frac{1}{nK} \sum_{i=1}^{n} \sum_{l=1}^{K} (|x_{i,j} - x_{i,j}^{M_l}| - |x_{i,j} - x_{i,j}^{H_l}|) \ , \tag{5.1}$$

where $x_{i,j}$, $x_{i,j}^{M_l}$, or $x_{i,j}^{H_l}$ denotes the jth component of instance $\mathbf{x}_i$, its lth closest Miss $\mathbf{x}_i^{M_l}$, or its lth closest Hit $\mathbf{x}_i^{H_l}$, respectively. n is the total number of instances, and K is the number of Hits or Misses considered for each instance. We used Weka's implementation of ReliefF with the default setting $K = 10$.

ReliefF appears similar to our proposed instance weighting algorithm since both algorithms involve distance calculation between an instance and its Hits or Misses along each feature for all instances. However, the two algorithms are intrinsically different. ReliefF is a feature weighting algorithm; it produces *feature* weights according to Eq. (5.1) which does not explicitly construct the margin vector for each instance, but takes an average of the margins over all instances. Our instance weighting algorithm produces *instance* weights by explicitly projecting each instance to its margin vector in the margin vector feature space (based on Eq. (4.2)) and a successive instance weighting procedure in the margin vector feature space. Our instance weighting algorithm can be used as a preprocessing step for any feature selection algorithms which can be extended to incorporate instance weights.

5.2.2 Instance Weighting SVM-RFE and Instance Weighting ReliefF

Given a training set, SVM-RFE and ReliefF select features based on the original training set where every instance is equally weighted. They can be extended to work on a weighted training set produced by our instance weighting algorithm. We refer to this version of SVM-RFE or ReliefF as instance weighting (IW) SVM-RFE or instance weighting (IW) ReliefF, respectively. We next explain how instance weights are incorporated into each algorithm.

For SVM-RFE, feature weights are determined based on the final chosen hyperplane of soft-margin SVM which is decided by the trade-off between maximizing the margin and minimizing the training error [Cortes and Vapnik 1995]. With an instance weight $w_i > 0$ assigned to each instance, the original objective function of soft-margin SVM is extended as follows:

$$\min \quad \frac{1}{2}\|\mathrm{w}\|^2 + C\sum_{i=1}^{n} w_i\xi_i \ , \tag{5.2}$$

where the first component is the opposite of the margin, and ξ_i (value of the slack variable) and C in the second component respectively capture the error of each instance caused by the hyperplane and the error penalty. For instances with $\xi_i > 0$, increased or decreased instance weight influences the error term (and hence the chosen hyperplane) by amplifying or reducing the effect of ξ_i. When all instances have equal weight, Eq. (5.2) becomes the original objective function of soft-margin SVM.

To see how the extension in Eq. (5.2) affects the variance of feature weighting by SVM, let us revisit the example in Figure 4.2. Without instance weighting, the optimal decision boundary is sensitive to the presence or absence of the two support vectors highlighted by the solid circles (assuming a reasonably large penalty term C). With instance weighting, these two instances will receive much lower weights than the rest of the instances. Given the same penalty term C, the optimal decision boundary

will be determined by alternative support vectors which lie much closer to the center of all instances in the margin vector feature space, and hence less sensitive to training data variations.

For ReliefF, feature weights are determined based on the weighting function in Eq. (5.1). With an instance weight $w_i > 0$ assigned to each instance, the original weighting function is extended as follows:

$$W(X_j) = \sum_{i=1}^{n} w_i \sum_{l=1}^{K} (w_i^{M_l} |x_{i,j} - x_{i,j}^{M_l}| - w_i^{H_l} |x_{i,j} - x_{i,j}^{H_l}|) \tag{5.3}$$

where w_i, $w_i^{M_l}$, or $w_i^{H_l}$ denotes the weight of instance $\mathbf{x}_i$, its lth closest Miss $\mathbf{x}_i^{M_l}$, or its lth closest Hit $\mathbf{x}_i^{H_l}$, respectively. Intuitively, instances with higher weights will have bigger influence on deciding the feature weights, and vice versa. Eq. (5.3) becomes the original weighting function Eq. (5.1) when all instances have equal weight $1/n$, and all Hits and Misses have equal weight $1/K$.

5.2.3 Ensemble SVM-RFE and Ensemble ReliefF

Given a training set, the bagging ensemble framework [Saeys et al. 2008] first generates a number of bootstrapped training sets, and then repeatedly applies a base feature selection algorithm (e.g., SVM-RFE or ReliefF) on each of the newly created training sets to generate a number of feature rankings. To aggregate the different rankings into a final consensus ranking, the complete linear aggregation scheme sums the ranks of a feature decided based on all bootstrapped training sets. We refer to the ensemble version of SVM-RFE or ReliefF as ensemble SVM-RFE or ensemble ReliefF, respectively. In our implementation, we use 20 bootstrapped training sets to construct each ensemble.

5.3 Evaluation Measures

5.3.1 Subset Stability Measure

The variance defined in Chapter 3 naturally quantifies the instability of a feature selection algorithm w.r.t. feature weights. The stability of a feature selection algorithm can also be measured w.r.t. the selected subsets. Following [Kalousis et al. 2007] and [Loscalzo et al. 2009], we take a similarity based approach where the stability of a feature selection algorithm is measured by the average over all pairwise similarity comparisons among all feature subsets obtained by the same algorithm from different subsamplings of a data set. Let $\{D_i\}_{i=1}^{q}$ be a set of subsamplings of a data set of the same size, and S_i be the subset selected by a feature selection algorithm $\mathcal{F}$ on the subsampling D_i. The stability of $\mathcal{F}$ is given by:

$$\overline{Sim} = \frac{2\sum_{i=1}^{q-1}\sum_{j=i+1}^{q} Sim(S_i, S_j)}{q(q-1)} , \tag{5.4}$$

where $Sim(S_i, S_j)$ represents a similarity measure between two subsets.

The stability of $\mathcal{F}$ depends on the specific choice of the similarity measure $Sim(S_i, S_j)$. Simple measures such as the percentage of overlap or Jaccard index can be applied as in [Kalousis et al. 2007]. These measures tend to produce higher values for larger subsets due to the increased bias of selecting overlapping features by chance. To correct this bias, [Kuncheva 2007] suggested the use of the Kuncheva index, defined as follows:

$$Sim(S_i, S_j) = \frac{|S_i \cap S_j| - (k^2/d)}{k - (k^2/d)} , \tag{5.5}$$

where d denotes the total number of features in a data set, and $k = |S_i| = |S_j|$ denotes the size of the selected subsets. The Kuncheva index takes values in [-1,1], with larger value indicating larger number of common features in both subsets. The k^2/d term corrects a bias due to the chance of selecting common features between two randomly

chosen subsets. An index close to zero reflects that the overlap between two subsets is mostly due to chance.

The Kuncheva index only considers overlapping genes between two gene subsets, without taking into account non-overlapping but highly correlated genes which may correspond to coordinated molecular changes. To address this issue, Zhang et al. proposed a measure called percentage of overlapping genes-related, POGR, defined as follows [Zhang et al. 2009]:

$$POGR(\mathbf{r}_i, \mathbf{r}_j) = \frac{r + O_{i,j}}{k_i} \ , \tag{5.6}$$

where $k_i = |\mathbf{r}_i|$ denotes the size of the gene subset $\mathbf{r}_i$, $r = |\mathbf{r}_i \cap \mathbf{r}_j|$ denotes the number of overlapping genes, and $O_{i,j}$ denotes the number of genes in $\mathbf{r}_i$ which are not shared but significantly positively correlated with at least one gene in $\mathbf{r}_j$. To normalize the bias effect of subset size, nPOGR, the normalized POGR, is defined as:

$$nPOGR(\mathbf{r}_i, \mathbf{r}_j) = \frac{r + O_{i,j} - E(r) - E(O_{i,j})}{k_i - E(r) - E(O_{i,j})} \ , \tag{5.7}$$

where $E(r)$ is the expected number of overlapping genes, and $E(O_{i,j})$ is the expected number of genes in $\mathbf{r}_i$ which are not shared but significantly positively correlated with at least one gene in $\mathbf{r}_j$, for two gene subsets (with size $|\mathbf{r}_i|$ and $|\mathbf{r}_j|$) randomly extracted from a given data set. The term $E(r)$ or $E(O_{i,j})$ respectively corrects the bias due to the chance of selecting common genes or significantly correlated genes between two randomly chosen gene subsets. Both definitions of POGR and nPOGR are nonsymmetric because it is possible that $|\mathbf{r}_i| \neq |\mathbf{r}_j|$ and/or $O_{i,j} \neq O_{j,i}$.

In our stability study, since we are interested in pairwise similarity between a number of gene subsets of equal size, we extend the original nPOGR measure into a symmetric measure by combining $nPOGR(\mathbf{r}_i, \mathbf{r}_j)$ and $nPOGR(\mathbf{r}_j, \mathbf{r}_i)$ as follows:

$$nPOGR(\mathbf{r}_i, \mathbf{r}_j) = \frac{r + O - E(r) - E(O)}{k - E(r) - E(O)} \ , \tag{5.8}$$

where $O = (O_{i,j} + O_{j,i})/2$, and $E(O)$ is the expected number of genes in one gene subset which are not shared but significantly positively correlated with at least one gene in the other subset, for any pair of gene subsets (with the same size) randomly extracted from a given data set. Note that this measure becomes the Kuncheva index if the two terms O and $E(O)$ about significantly correlated genes are removed. According to [Zhang et al. 2009], $E(O)$ is estimated based on 10,000 randomly generated pairs of gene subsets. Significantly correlated genes are determined based on Pearson correlation with 0.1% FDR control.

5.3.2 Predictive Performance Measures

The predictive accuracy is measured by the percentage of correctly classified instances over the overall instances. In our empirical study, we adopt Cross-Validation(CV) accuracy as implemented in Weka's implementation as our base measure for evaluating the classification performance, which is thoroughly computed for both synthetic data and real-world data.

Since the real-world data sets used in this study contain imbalanced class distributions (in particular, the lung cancer data set), we also adopt a commonly used measure in this context, the area under the receiver operating characteristic (ROC) curve (denoted as AUC), to compare the classification performance of different methods. AUC is a function of two class-specific measures: sensitivity and specificity, defined, respectively as the proportion of correctly classified samples in the positive and the negative classes. We adopted Weka's implementation of calcualting the area under ROC curve, which uses Mann Whitney statistic essentially.

6 Experiments on Synthetic Data

This chapter extensively verifies the theoretical and empirical frameworks based on synthetic data sets and a popular feature selection algorithm SVM-RFE. Section 6.1 describes the details on how to generate the desired synthetic data and experiment settings. Section 6.2 evaluate the bias-variance decomposition of feature selection error and variance reduction via instance weighting with respect to feature weights. Section 6.3 compares the stability and predictive performance with respect to the selected subsets. Section 6.4 evaluates the sample size effects on feature selection and instance weighting.

6.1 Experimental Setup

The data distribution used to generate training and test sets consists of 1000 random variables (features) from a mixture of two multivariate normal distributions: $\mathcal{N}_1(\mu_1, \Sigma)$ and $\mathcal{N}_2(\mu_2, \Sigma)$, with means

$$\mu_1 = (\underbrace{0.5, ..., 0.5}_{50}, \underbrace{0, ..., 0}_{950}) \ ,$$

$$\mu_2 = (\underbrace{-0.5, ..., -0.5}_{50}, \underbrace{0, ..., 0}_{950}) \ ,$$

and covariance

$$\Sigma = \begin{bmatrix} \Sigma_1 & 0 & \cdots & 0 \\ 0 & \Sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_{100} \end{bmatrix} ,$$

where Σ is a block diagonal matrix, and Σ_i is a 10×10 square matrix with elements 1 along its diagonal and 0.8 off its diagonal. So, there are 100 correlated groups with 10 features per group. The class label of each instance from this distribution is decided by the sign of a linear combination of all feature values according to the optimal weight vector

$$\mathbf{r}^* = (\underbrace{0.02, ..., 0.02}_{50}, \underbrace{0, ..., 0}_{950}) .$$

Note that the weights of all features sums to 1. Features in the first 5 groups are equally relevant, and features in the other groups are irrelevant.

To measure the variance, bias, and error of a given feature selection algorithm according to the definitions in Chapter 3, we simulate $\mathscr{D}$, the distribution of all possible training sets, by 500 training sets randomly drawn from the above data distribution. Each training set consists of 100 instances with 50 from $\mathcal{N}_1$ and 50 from $\mathcal{N}_2$. To measure the predictive performance of the selected features, we also randomly draw a test set of 5000 instances.

For experiments with synthetic data, we focus on the SVM-RFE algorithm. To measure the variance, bias, and error of SVM-RFE, we alternatively view the RFE process as an iterative feature weighting process, and associate a normalized weight vector $\mathbf{r} = (r_1, ..., r_d)$ $(\sum_{j=1}^{d} r_j = 1, r_j \geq 0)$ to the full set of d features. At each iteration of the RFE process, the weight of each feature is determined according to $r_j = \frac{|w_j|}{\sum_{j=1}^{d} |w_j|}$, where $w_j = 0$ for the eliminated features, and w_j equals the weight of feature j in the current optimal hyperplane for the remaining features.

6.2 Bias-Variance Decomposition and Variance Reduction w.r.t. Feature Weights

Given the 500 training sets described above, SVM-RFE is applied on each training set, and the resulting normalized weights for all features are recorded at each iteration of the RFE process. The variance, bias, and error over all features (as defined in Eqs. (3.1.1)-(3.4)) are then calculated at each iteration. To verify the effect of instance weighting on variance reduction, the proposed instance weighting algorithm is also applied on each training set to produce its weighted version. SVM-RFE is then repeatedly applied on the 500 weighted training sets in order to measure the variance, bias, and error for **IW** SVM-RFE.

Figure 6.1 reports the variance, bias, and error of SVM-RFE and **IW** SVM-RFE across the RFE process (until 10 features remain at the 40th iteration). We can observe three major trends from the figures which verify the theoretical framewrok we proposed in Chapter 3.

First, for both versions of SVM-RFE, at any iteration, the error is always equal to the sum of the variance and the bias, which is consistent with the bias-variance decomposition of error shown in Eq. (3.4). Second, for both versions of SVM-RFE, the error is first dominated by the bias during the early iterations when many irrelevant features are assigned non-zero weights, and then becomes dominated by the variance during the later iterations when some relevant features are assigned zero weights. In particular, the error of **IW** SVM-RFE reaches to almost zero at the 28th iteration when the number of remaining features is closest to 50 (the number of truly relevant features). Before or after that point, its error almost solely results from its bias or variance, respectively. Third, **IW** SVM-RFE exhibits significantly lower variance and bias (hence, lower error) than SVM-RFE when the number of remaining features approaches to 50.

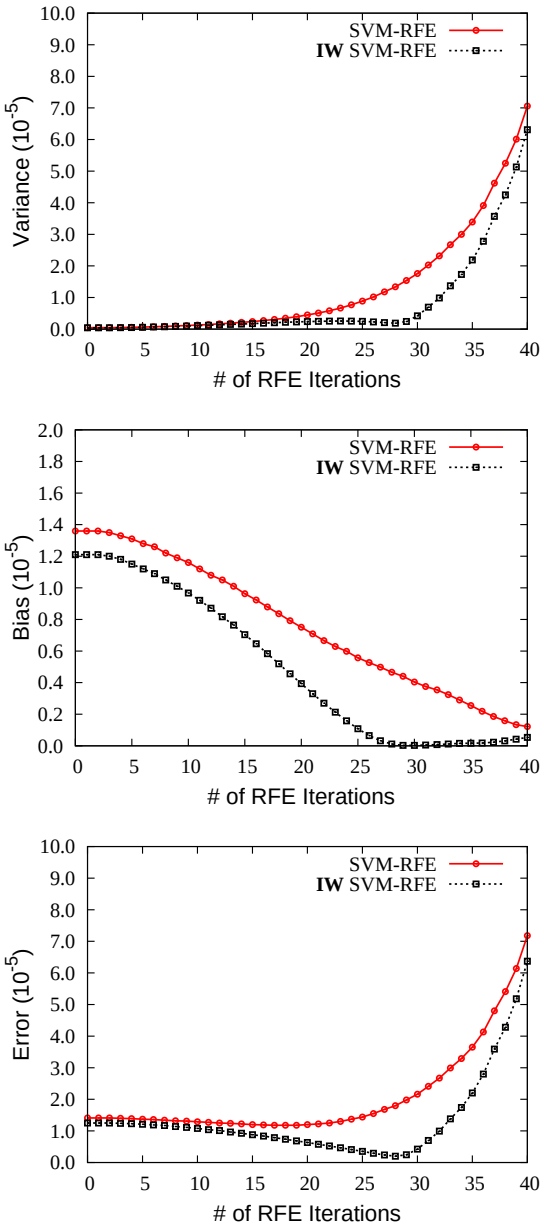

Figure 6.1. Variance, Bias, and Error of the feature weights assigned by the conventional and Instance Weighting (**IW**) versions of the SVM-RFE algorithm at each iteration of the RFE process for synthetic data.

6.3 Stability and Predictive Performance w.r.t. Selected Subsets

We next study the impacts of variance reduction on the stability and the predictive performance of the selected subsets. Figure 6.2 (upper) compares the subset stability

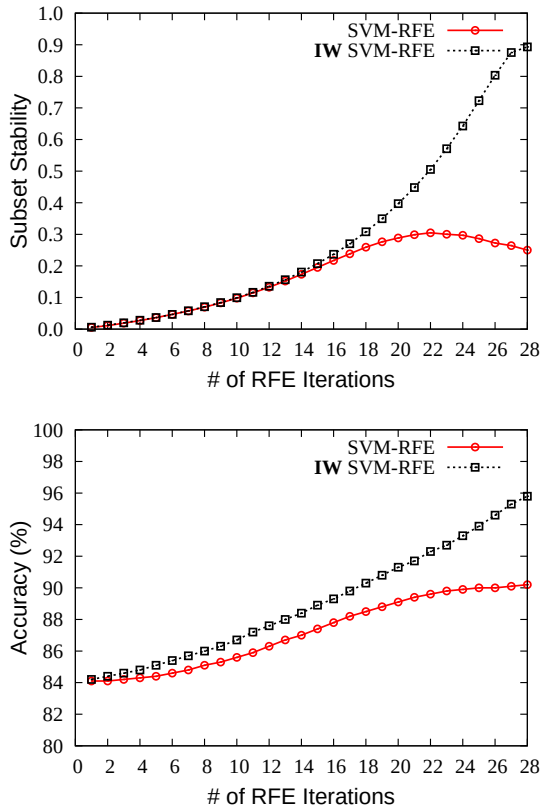

Figure 6.2. Stability (by Kuncheva Index) and predictive performance (by accuracy of linear SVM) of the selected subsets by the conventional and Instance Weighting (**IW**) versions of the SVM-RFE algorithm at each iteration of the RFE process for synthetic data.

(by Kuncheva index) of SVM-RFE and **IW** SVM-RFE across the RFE process (until about 50 features remain at the 28th iteration). To measure predictive performance, for each training set, a linear SVM classifier is trained based on the selected subset at each RFE iteration and tested on the independent test set. Figure 6.2 (lower) compares the average predictive accuracy (over the 500 training/test trials) of linear SVM at each RFE iteration.

From Figure 6.2 (upper), we can observe that the stability of the subsets selected by **IW** SVM-RFE becomes significantly higher than those selected by SVM-RFE as the number of selected features approaches to the number of truly relevant features at the 28th iteration. Examining the trend of subset stability together with the trend of

variance (in Figure 6.1), we can see that the reduction of variance by instance weighting goes in parallel with the improvement of subset stability, except for the early iterations when irrelevant features are eliminated largely by chance. Note that both versions of SVM-RFE exhibit very low stability during the early iterations, because of the inclusion of the correction term in the Kuncheva index. From Figure 6.2 (lower), we can observe that the subsets selected by **IW** SVM-RFE also result in higher predictive accuracy than those selected by SVM-RFE. The difference is particularly significant during iterations when **IW** SVM-RFE exhibits much higher stability than SVM-RFE. Overall, results from Figures 6.1 and 6.2 demonstrate that variance reduction by instance weighting, an approach for a better bias-variance tradeoff, can lead to increased subset stability as well as improved predictive accuracy based on the selected features.

6.4 Sample Size Effects on Feature Selection and Instance Weighting

Results reported so far are based on the synthetic data distribution described in Section 6.1, with 100 training instances in each training set. We now study the effects of training sample size on SVM-RFE and **IW** SVM-RFE. We first look into how increasing sample size affects the variance, bias, and error of the two versions of SVM-RFE. Instead of examining the three metrics across the entire RFE process, we focus on the iteration when 50 features are selected. As shown in Figure 6.3, the performance of SVM-RFE clearly depends on the sample size. Both the variance and the bias (hence the error) of SVM-RFE consistently decrease as the sample size increases from 50 to 1000. In contrast, the performance of **IW** SVM-RFE exhibits a much less dependency on the sample size. Its three metrics quickly converge to near zero when the sample size approaches to 200. Matching trends can be observed from Figure 6.4 which depicts the stability and the predictive performance of the selected

subsets by the two algorithms under increasing sample size. Both subset stability and predictive performance of SVM-RFE consistently improve with the increase of sample size, while the two metrics of **IW** SVM-RFE converge to almost perfect values as soon as the sample size reaches 200.

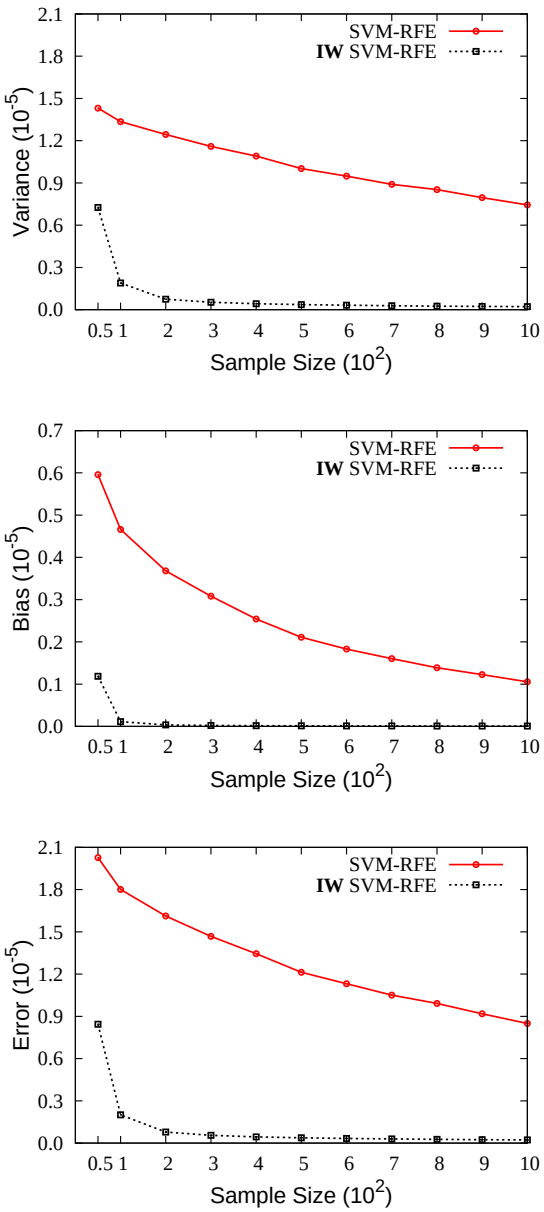

Figure 6.3. Variance, Bias, and Error of the feature weights assigned by the conventional and Instance Weighting (**IW**) versions of the SVM-RFE algorithm when 50 features selected on synthetic data with increasing sample size.

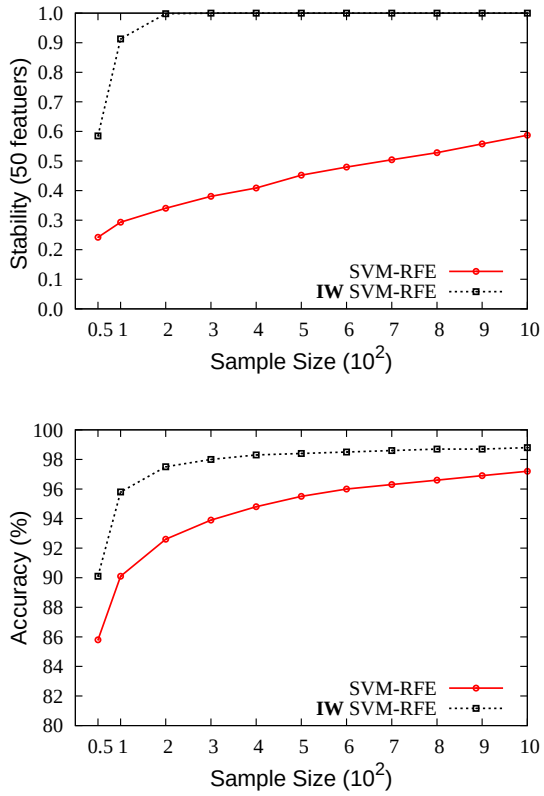

Figure 6.4. Stability (by Kuncheva Index) and predictive performance (by accuracy of linear SVM) of the selected subsets by the conventional and Instance Weighting (**IW**) versions of the SVM-RFE algorithm when 50 features selected on synthetic data with increasing sample size.

Overall, results from Figures 6.3 and 6.4 demonstrate: (i) the sample size dependency of the performance of feature selection algorithms such as SVM-RFE and (ii) the effectiveness of instance weighting on alleviating such dependency. It is worthy of noting that the performance of **IW** SVM-RFE at sample size 200 is significantly better than that of SVM-RFE at sample size 1000. Such observation indicates that the proposed instance weighting framework is an effective alternative to increasing sample size for improving the stability and predictive performance of feature selection algorithms. Recall that in many real-world applications like microarray data analysis, increasing the number of training samples could be impractical or very costly.

7 Experiments on Real-World Data

In this chapter, we further verifies that the empirical framework is effective at reducing the variance and improving the subset stability of two representative feature selection algorithms, SVM-RFE and ReliefF, while maintaining comparable predictive accuracy based on the selected features by thoroughly experimenting with the real-world microarray data sets under different experiment settings and various evaluation measures for both stability and classification performance. The proposed instance weighting framework is also shown to be more effective and efficient than the ensemble framework in terms of stability and time complexity. Section 7.1 describes the real-world data sets used in our study and experiment settings. Section 7.2 verifies the variance reduction via instance weighting scheme with respect to the feature weights. Section 7.3 shows the stability and predictive performance results with respect to the selected subsets. Section 7.4 further evaluate the stability of feature selection through consensus gene signatures. Section 7.5 compares the algorithm efficiency of different algorithms.

7.1 Experimental Setup

We experimented with four frequently studied microarray data sets characterized in Table 7.1. For the Lung data set, we applied a t-test to the original data set and only kept the top 5000 features in order to make the experiments more manageable.

Table 7.1. Summary of microarray data sets.

Data Set	# Features	# Instances	Source
Colon	2000	62	[Alon et al. 1999]
Leukemia	7129	72	[Golub et al. 1999]
Prostate	6034	102	[Singh et al. 2002]
Lung	12533	181	[Gordon et al. 2002]

In this section, we verify the effectiveness of the instance weighting framework for both SVM-RFE and ReliefF algorithms. Furthermore, we compare the instance weighting framework with the ensemble feature selection framework. To do so, we evaluate the performance of the original, ensemble, and instance weighting (IW) versions of SVM-RFE and ReliefF on each data set, using two different validation procedures separately.

One is the 10×10 folds cross-validation (CV) procedure. Given a data set and a random 10-fold partition of it, the three versions of SVM-RFE and ReliefF are repeatedly applied to 9 out of the 10 folds to produce feature weights and select subsets of features at various sizes, while a different fold is hold out each time. For each selected subset, both a linear SVM and a KNN (K=1) classifiers are trained based on the selected features and the training set, and then tested on the corresponding hold-out test set. The variance and subset stability of each algorithm is measured in the same way as for synthetic data. The predictive performance of each algorithm is measured based on the CV accuracies of the linear SVM and KNN classifiers. This entire procedure is repeated 10 times, and the average over 10 runs is reported for each performance metric.

The other one is a random repetition procedure. For each data set used in the study, the entire data set was randomly split into the training set and the test set, with 2/3 of all the samples of each class in the training set, and the rest in the test set. The original, ensemble, and instance weighting versions of a baseline algorithm (SVM-RFE or ReliefF) were applied on the training set to select subsets of genes

at various sizes. It's noteworthy that the subset size of selected features varies from 10 to 200, which is also different from the setting from previous procedure.For each selected subset, both a linear SVM classifier (with default C parameter in Weka) and a K-nearest neighbor classifier (K=1) were trained based on the selected genes and the training set, and then tested on the corresponding test set. For each data set, the above procedures were repeated 100 times. The stability of a selection method was measured over the 100 subsamplings of the data set according to Eq. (5.4). The classification performance of the method was measured by the average AUC over the 100 random training/test splits.

Apparently, these two experiment procedures differ in many aspects, such as training and test sample size, number of repetitions, subset size of selected features, evaluations metrics and so on. Therefore, it provide a comprehensive evaluation of the framework of stable feature selection and the results would be more convincing.

7.2 Variance Reduction w.r.t. Feature Weights

Since the true relevance of features is usually unknown for real-world data, it is infeasible to measure the bias and error of feature selection and thus we can not verify the variance-bias decomposition of feature selection error for real-world data. Nevertheless, we can still evaluate the effect of instance weighting on the variance of SVM-RFE following a similar procedure as used for synthetic data.

Figure 7.1 (a)-(d) report the variance of SVM-RFE and **IW** SVM-RFE across the RFE process for each of the four microarray data sets. Since these data sets contain various numbers of features, to make all figures comparable along the horizontal axis, each variance curve is made to show 40 iterations starting from when about 1000 features remain until when about 10 features remain (the same range shown in Figure 6.1 for synthetic data). As shown from Figure 7.1, the variance for both versions of SVM-RFE remains almost zero in the early iterations. However, in the later

iterations, the variance of SVM-RFE increases sharply as the number of remaining features approaches to 10, while the variance of **IW** SVM-RFE shows a significantly slower rate of increase than SVM-RFE. Such observations demonstrate the effect of instance weighting on variance reduction on real-world data.

The variance curves for SVM-RFE and **IW** SVM-RFE appear to be indistinguishable during the early iterations because the variance values in the beginning are several orders of magnitude smaller than those at the end of the RFE process. To illustrate the significant difference of performance between the two algorithms, we provide a zoomed-in view on the variance of SVM-RFE and **IW** SVM-RFE at the first indexed iteration for each data set in sub-graph (e), where an error bar shows the standard deviation over the 10 random runs.The variance score for each data set is re-scaled for clearer presentation. Clearly, instance weighting significantly reduces the variance on all data sets (except Leukemia) even in the beginning of the RFE process, in spite of the small magnitude of the variance values (due to the normalization applied to feature weight vectors). For the iterations close to the end of the RFE process, the error bars remain much smaller compared to the gap between the two curves. In order to have a clear presentation, the error bars are omitted from sub-figures (a)-(d).

We now examine the effectiveness of instance weighting on variance reduction for ReliefF. Note that ReliefF does not involve an RFE procedure used in SVM-RFE. Figure 7.2 reports the variance of ReliefF and **IW** ReliefF for all four data sets. Except Prostate, instance weighting significantly reduces the variance of ReliefF. Results in this section demonstrate the effect of instance weighting on the variance of two representative algorithms based on real-world data. We next examine how it impacts feature subset stability and predictive performance of these algorithms.

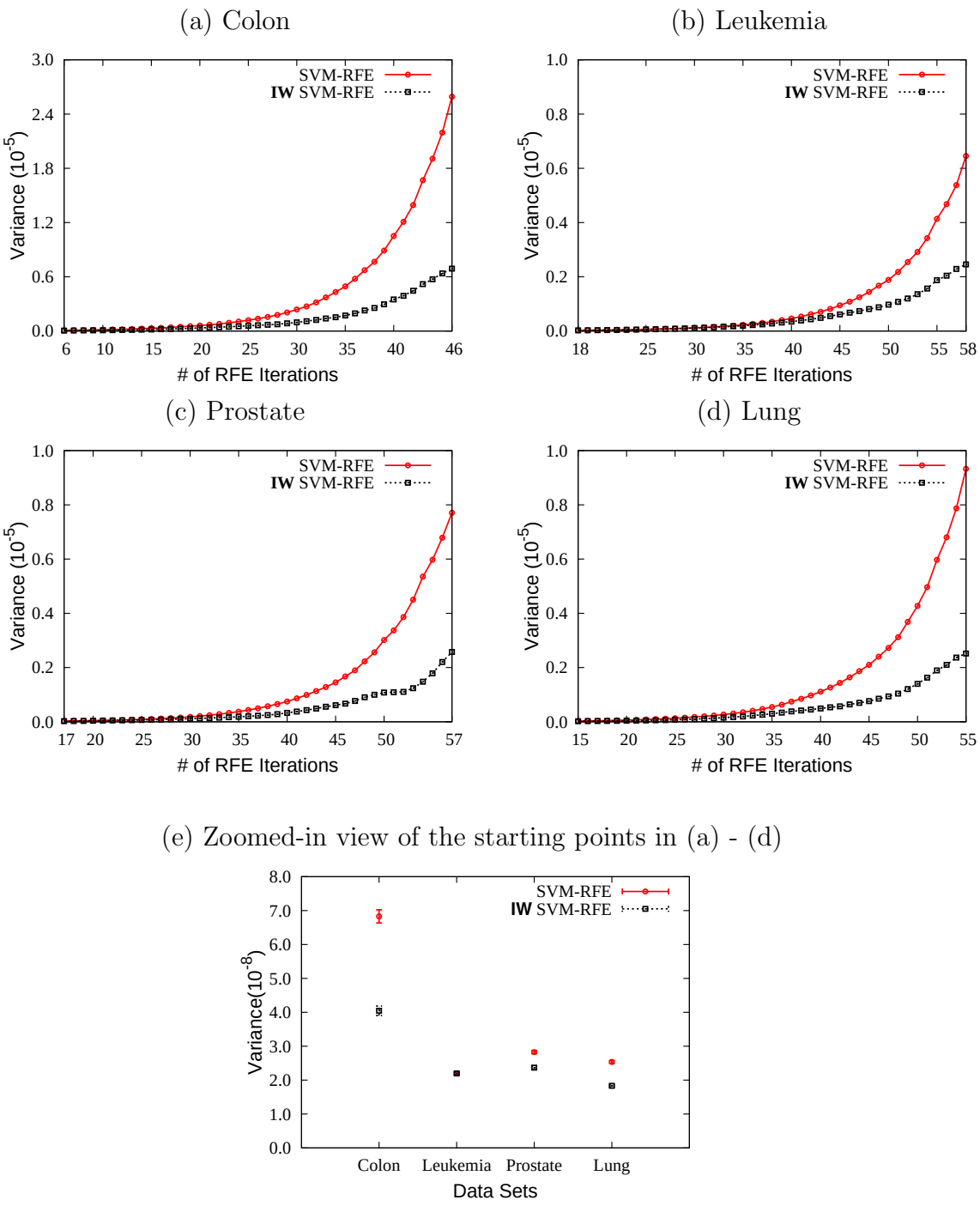

Figure 7.1. Variance of the feature weights assigned by the conventional and Instance Weighting (**IW**) versions of the SVM-RFE algorithm at each iteration of the RFE process for four data sets (a)-(d). A zoomed-in view of the starting points in each figure is shown in (e), where an error bar shows the standard deviation over 10 runs.

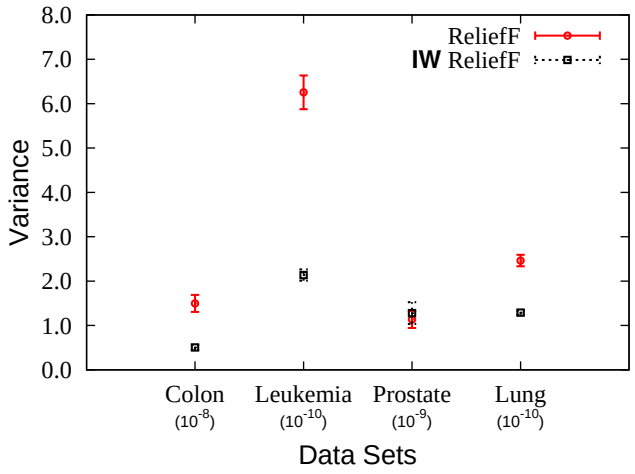

Figure 7.2. Variance of the feature weights assigned by the conventional and Instance Weighting (**IW**) versions of the ReliefF algorithm for four data sets. Scales of variance values are shown in parentheses.

7.3 Stability and Predictive Performance w.r.t. Selected Subsets

7.3.1 Stability w.r.t. Selected Subsets

7.3.1.1 Cross-validation Procedure with Kuncheva Index

Figure 7.3 reports the subset stability measured by Kuncheva Index under the setting of 10-time-10-fold cross-validation across different numbers of selected features(from 10 to 50 with an interval of 10) for the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of SVM-RFE on four data sets. Instance weighting significantly improves the stability of SVM-RFE, which is consistent with the variance reduction effect of instance weighting observed above. Moreover, a comparison of the stability of **IW** SVM-RFE and **En** SVM-RFE indicates that instance weighting is more effective than ensemble for improving the stability of SVM-RFE.

Figure 7.4 reports the stability results for ReliefF in three versions on four data sets. Comparing Figure 7.4 with Figure 7.3, we can observe that the stability of

ReliefF is consistently higher than SVM-RFE under the same stability measure. Although ReliefF shows relatively more stable results, instance weighting in general improves its stability for all data sets. It is worth mentioning that for the Lung data set, **IW** ReliefF exhibits almost perfect subset stability. In contrast to instance weighting, the ensemble method exhibits a negative effect on the stability of ReliefF.

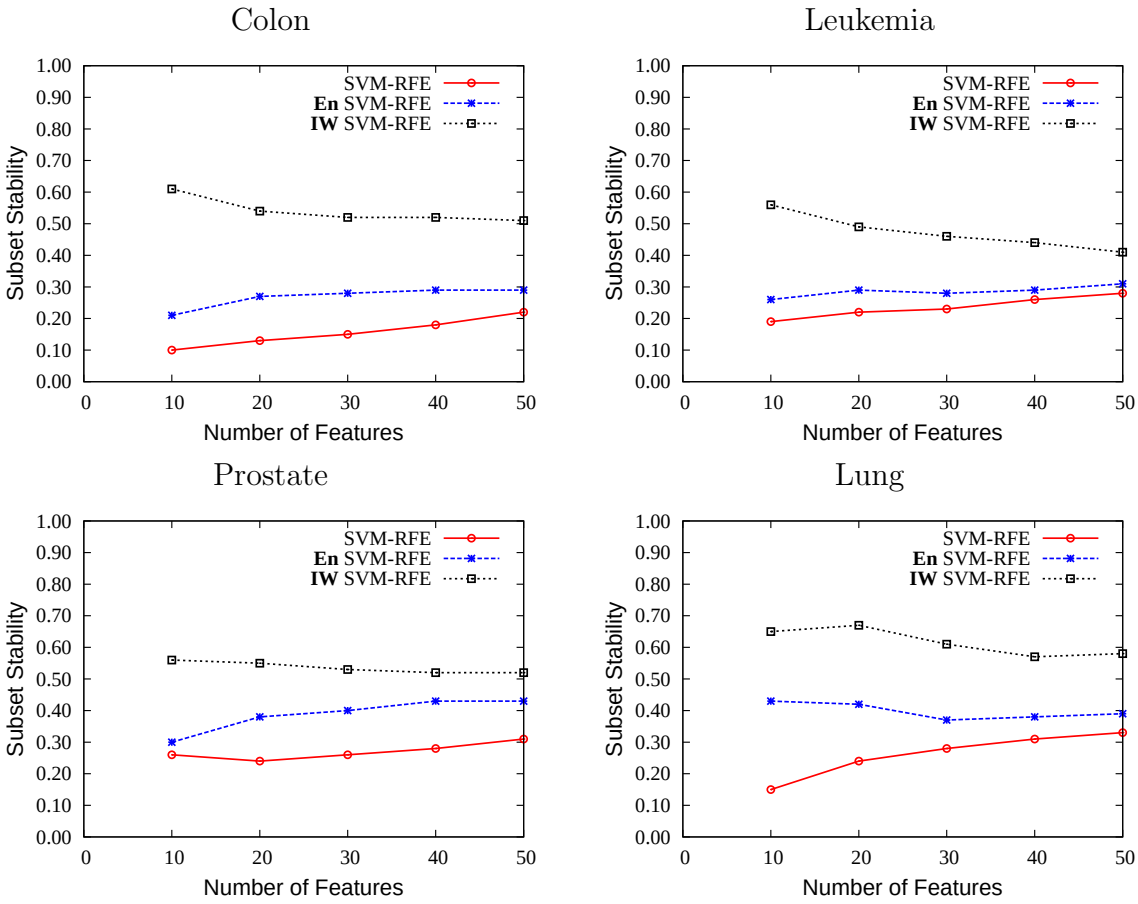

Figure 7.3. Stability (by Kuncheva Index) of the selected subsets by the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of the SVM-RFE algorithm for four data sets.

7.3.1.2 Random Repetition Procedure with nPOGR

Figure 7.5 reports the subset stability measured by nPOGR under the setting of 100 random repetitions across different numbers of selected features(from 10 to 200 with

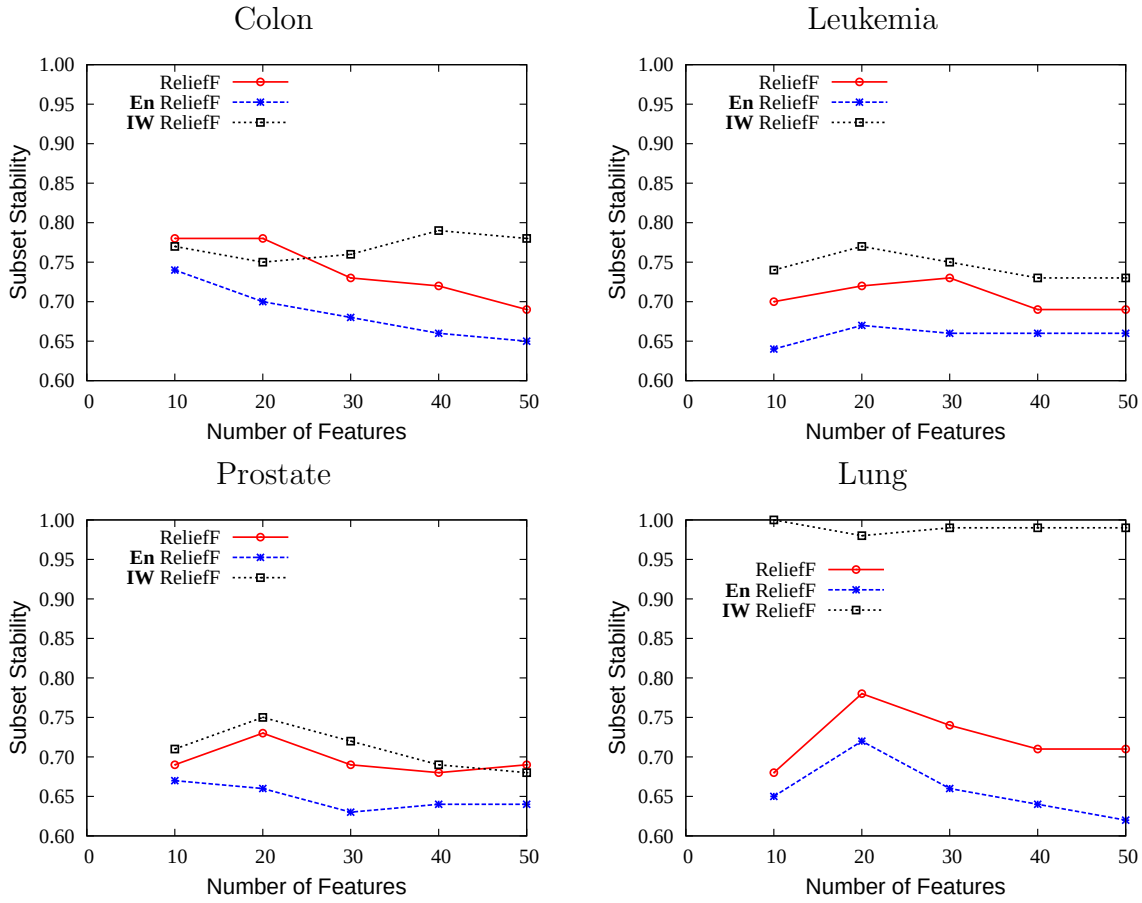

Figure 7.4. Stability (by Kuncheva Index) of the selected subsets by the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of the ReliefF algorithm for four data sets.

an interval of 50) for the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of SVM-RFE on four data sets. Instance weighting significantly improves the stability of SVM-RFE and is more effective than ensemble for improving the stability of SVM-RFE. This is consistent with the stability trend measured by Kunchve index as above.

Figure 7.6 reports the stability results for ReliefF in three versions on four data sets. Instance weighting significantly improves the stability of ReliefF for all data sets. Compared to **En** ReliefF, the stability of **IW** ReliefF is also consistently better except for Prostate data set.

7.3.1.3 Overall Conclusions

By comparing the two sets of figures above, we have several observations. First, the stability of ReiefF is consistently higher SVM-RFE no matter which evaluation measure is used. Second, instance weighting can significantly improves the stability of traditional feature selection algorithms and also more effective than the ensemble approach under both experiment setting by either of the two evaluation measure.

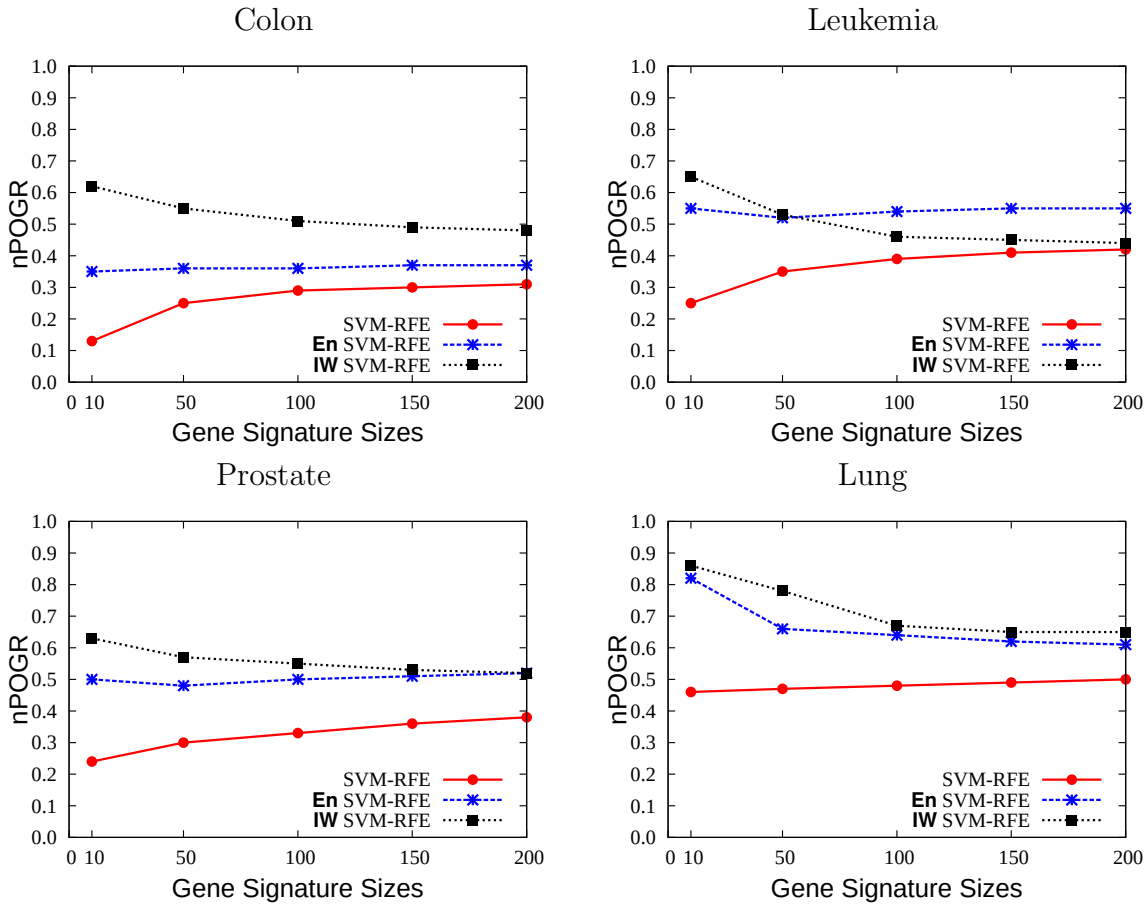

Figure 7.5. Stability (by nPOGR) of the selected subsets by the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of the SVM-RFE algorithm for four data sets.

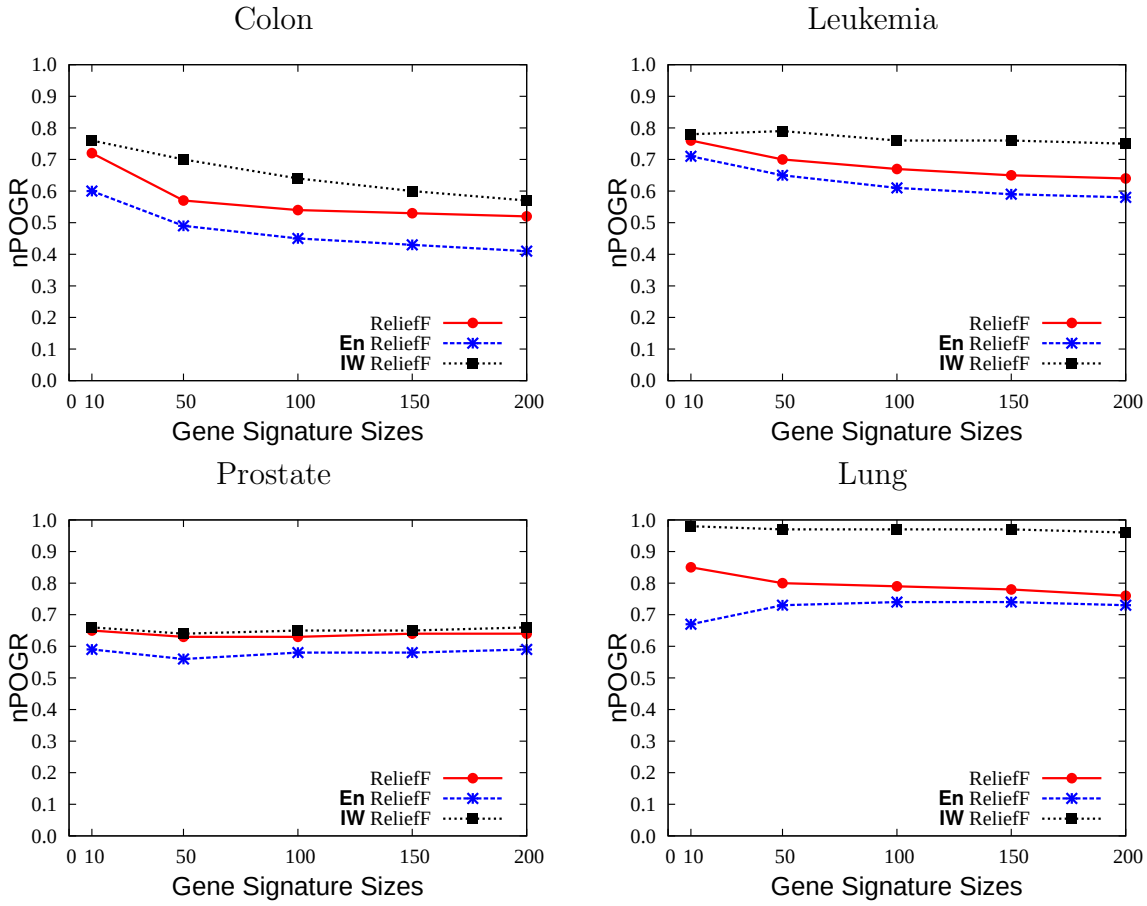

Figure 7.6. Stability (by nPOGR Index) of the selected subsets by the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of the ReliefF algorithm for four data sets.

7.3.2 Predictive Performance

7.3.2.1 Cross-validation Procedure with CV Accuracy

Tables 7.2 and 7.3 report the CV accuracy (average value ± standard deviation) of linear SVM and 1NN based on the selected features(from 10 to 50 with an interval of 10) by the three versions of SVM-RFE and ReliefF under the 10-time-10-fold cross-validation setting, respectively. For both SVM-RFE and ReliefF, the three versions in general lead to very similar predictive accuracy. Except for a few cases, the differences in the average accuracy values produced by the three versions are insignificant given

Table 7.2. Classification performance measured by the CV accuracy(average value ± standard deviation) of the linear SVM and 1NN for the Conventional, Ensemble (**En**), and Instance Weighting versions (**IW**) of SVM-RFE at increasing gene signature sizes for four data sets.

			Number of Selected Features				
Data Set	Classifier	Selection Method	10	20	30	40	50
Colon	SVM	SVM-RFE	82.1±3.5	82.1±3.8	81.9±4.9	82.4±3.6	82.1±3.3
		En SVM-RFE	82.1±4.5	83.9±2.8	83.2±4.5	82.5±4.0	83.2±4.0
		IW SVM-RFE	82.8±2.3	86.6±1.3	86.3±3.1	85.6±3.6	84.6±2.6
	1NN	SVM-RFE	76.8±3.8	78.7±3.9	79.5±3.7	81.0±3.0	81.8±3.5
		En SVM-RFE	76.5±4.5	80.3±2.5	79.0±3.1	79.2±3.1	80.2±3.6
		IW SVM-RFE	76.4±4.0	77.6±5.6	77.7±3.3	78.8±2.4	79.7±2.6
Leukemia	SVM	SVM-RFE	95.0±2.0	96.0±1.4	96.7±1.2	96.8±0.7	97.1±0.8
		En SVM-RFE	94.4±1.3	96.0±1.0	96.2±0.9	95.8±1.1	96.8±1.3
		IW SVM-RFE	92.9±1.2	94.7±1.8	96.0±1.5	96.4±1.2	96.5±0.7
	1NN	SVM-RFE	93.6±2.2	95.3±1.2	95.8±1.6	95.7±1.0	96.5±1.8
		En SVM-RFE	93.3±1.9	94.2±2.2	95.1±1.8	95.4±3.0	95.7±2.4
		IW SVM-RFE	92.8±1.9	95.1±1.3	95.3±1.4	94.7±1.4	95.7±1.8
Prostate	SVM	SVM-RFE	91.9±2.3	92.3±2.0	93.0±1.6	92.6±1.6	93.8±0.9
		En SVM-RFE	93.0±2.5	92.9±1.3	93.8±1.9	94.4±1.7	94.1±1.2
		IW SVM-RFE	93.0±1.3	92.0±1.1	91.3±1.6	91.2±1.7	91.2±1.2
	1NN	SVM-RFE	90.3±3.1	90.5±3.9	91.7±2.7	91.7±2.5	92.3±1.2
		En SVM-RFE	89.7±2.7	91.7±2.7	91.6±2.3	92.1±2.2	92.3±1.8
		IW SVM-RFE	91.0±1.7	90.9±1.3	90.5±2.2	90.2±2.4	91.6±2.3
Lung	SVM	SVM-RFE	98.3±0.4	98.8±0.3	99.0±0.3	99.0±0.3	98.9±0.0
		En SVM-RFE	98.8±0.5	98.8±0.2	98.8±0.2	99.0±0.2	98.9±0.0
		IW SVM-RFE	98.5±0.5	98.8±0.2	98.9±0.0	99.1±0.3	99.0±0.2
	1NN	SVM-RFE	98.2±0.4	98.5±0.4	98.4±0.6	98.6±0.4	98.7±0.3
		En SVM-RFE	98.8±0.2	98.5±0.4	98.6±0.5	98.7±0.3	98.5±0.3
		IW SVM-RFE	98.8±0.6	98.5±0.6	98.8±0.2	98.9±0.0	98.9±0.0

the standard deviations. The accuracy results in Tables 7.2 and 7.3 verify that the increased stability resulted from instance weighting (as shown in Figures 7.3 and 7.4) is not at the price of accuracy.

7.3.2.2 Accuracy w.r.t AUC curve

Tables 7.4 and 7.5 report the the classification performance (by the AUC) (average value ± standard deviation) of linear SVM and 1NN based on the selected features(from 50 to 200 with an interval of 50) by the three versions of SVM-RFE and

Table 7.3. Classification performance measured by the CV accuracy (average value ± standard deviation) of the linear SVM and 1NN for the Conventional, Ensemble (**En**), and Instance Weighting versions (**IW**) of ReliefF at increasing gene signature sizes for four data sets.

			Number of Selected Features				
Data Set	Classifier	Selection Method	10	20	30	40	50
Colon	SVM	ReliefF	84.0±2.0	84.8±1.2	84.0±3.1	84.2±2.7	85.2±3.6
		En ReliefF	84.2±2.8	84.5±2.6	85.0±2.5	84.3±2.9	84.5±2.6
		IW ReliefF	83.9±2.6	83.7±2.2	84.4±1.3	84.2±2.5	83.9±4.3
	1NN	ReliefF	75.3±3.5	75.0±2.3	75.8±3.4	75.5±3.4	74.6±4.3
		En ReliefF	77.0±3.0	75.5±2.9	76.1±3.6	76.8±3.1	76.2±2.8
		IW ReliefF	74.7±4.6	76.9±3.4	75.7±2.6	75.8±3.2	76.1±2.6
Leukemia	SVM	ReliefF	92.1±1.5	94.7±1.1	95.1±1.3	96.0±1.2	96.0±0.8
		En ReliefF	92.6±0.9	93.9±1.3	94.4±1.6	95.3±1.5	95.4±1.5
		IW ReliefF	92.5±1.5	93.9±1.2	94.9±0.9	95.4±0.7	95.3±1.5
	1NN	ReliefF	91.9±1.4	91.9±1.7	91.7±2.1	92.4±2.3	91.9±2.6
		En ReliefF	91.8±1.8	91.9±3.0	91.1±1.8	91.7±2.5	91.8±2.1
		IW ReliefF	92.9±1.0	92.1±2.1	91.3±2.5	91.5±2.2	91.0±2.3
Prostate	SVM	ReliefF	92.3±1.1	92.9±1.4	92.1±1.6	92.4±1.8	91.2±1.3
		En ReliefF	92.1±1.4	92.5±2.3	91.7±1.1	92.0±1.1	91.4±1.7
		IW ReliefF	92.5±1.1	91.8±2.2	92.0±1.8	91.5±1.5	91.1±0.9
	1NN	ReliefF	88.3±2.2	87.5±2.5	86.9±2.8	87.3±2.1	87.2±2.6
		En ReliefF	86.4±2.9	86.5±2.3	85.5±3.1	85.7±2.0	86.4±2.0
		IW ReliefF	89.5±1.4	89.3±2.3	88.0±1.5	87.5±1.9	87.1±1.4
Lung	SVM	ReliefF	98.6±0.3	98.8±0.2	98.9±0.0	99.1±0.3	99.0±0.2
		En ReliefF	98.7±0.3	98.9±0.0	98.8±0.2	98.8±0.2	99.0±0.2
		IW ReliefF	98.6±0.5	98.8±0.2	98.9±0.0	98.8±0.2	98.8±0.2
	1NN	ReliefF	98.7±0.5	98.7±0.3	98.8±0.2	98.7±0.3	98.6±0.4
		En ReliefF	98.5±0.7	98.6±0.4	98.7±0.3	98.7±0.3	98.5±0.4
		IW ReliefF	98.8±0.6	98.7±0.4	98.5±0.5	98.6±0.3	98.5±0.5

ReliefF under the setting of 100 random repetitions, respectively. We can observe that the three versions of SVM-RFE (or ReliefF) in general perform very similar under each signature size. Although there are some marginal differences among the three versions in terms of the average AUC values at places, the differences are not statistically significant, given the large standard deviation values caused by the small sample size of the test sets.

Table 7.4. Classification performance measured by the AUC (average value ± standard deviation) of the linear SVM and 1NN for the Conventional, Ensemble (**En**), and Instance Weighting versions (**IW**) of SVM-RFE at increasing gene signature sizes for four data sets.

			Number of Selected Features				
Data	Classifier	Selection Method	10	50	100	150	200
Colon	SVM	SVM-RFE	76.4±9.5	77.5±8.2	79.2±8.7	79.4±8.5	80.1±8.7
		En SVM-RFE	80.3±7.9	79.4±9.0	78.6±8.3	78.6±9.1	79.4±8.7
		IW SVM-RFE	79.5±9.1	81.2±8.4	78.4±10.0	76.2±10.0	76.2±9.5
	1NN	SVM-RFE	73.0±9.6	76.6±8.9	78.4±8.7	78.1±8.4	77.8±8.5
		En SVM-RFE	73.6±8.7	77.1±9.1	79.6±8.4	80.1±8.8	79.6±8.0
		IW SVM-RFE	72.0±9.2	75.3±8.2	78.0±8.3	79.2±7.8	78.9±8.5
Leukemia	SVM	SVM-RFE	92.8±5.8	96.3±3.8	96.9±3.3	96.8±3.5	97.0±3.4
		En SVM-RFE	92.9±5.4	96.4±3.9	97.2±3.1	97.0±3.4	96.7±3.5
		IW SVM-RFE	91.2±5.6	96.2±3.9	96.4±3.3	96.5±3.4	96.8±3.5
	1NN	SVM-RFE	91.2±6.0	94.5±5.0	94.8±4.9	95.0±5.0	95.2±4.4
		En SVM-RFE	92.4±5.4	95.0±4.9	95.6±4.7	95.2±4.6	94.8±4.8
		IW SVM-RFE	90.7±5.3	94.1±4.9	95.4±4.4	95.4±4.5	95.4±4.4
Prost	SVM	SVM-RFE	89.8±5.1	91.3±4.1	92.1±3.8	92.1±4.3	92.2±3.9
		En SVM-RFE	92.9±4.1	92.0±4.5	92.0±4.6	92.6±4.0	92.7±4.3
		IW SVM-RFE	93.4±3.6	91.3±4.5	90.0±4.8	90.7±4.9	91.2±4.7
	1NN	SVM-RFE	88.2±4.9	90.5±4.6	90.8±4.4	91.1±4.2	91.5±4.5
		En SVM-RFE	89.6±4.7	91.4±4.6	90.8±4.4	90.7±4.4	90.7±4.2
		IW SVM-RFE	90.5±4.0	90.1±5.1	89.6±4.7	89.7±4.8	89.6±5.1
Lung	SVM	SVM-RFE	95.8±4.3	96.8±3.1	96.9±3.1	96.8±3.1	96.8±3.1
		En SVM-RFE	96.3±3.5	96.9±3.2	96.9±3.1	97.0±3.1	96.9±3.1
		IW SVM-RFE	94.7±4.7	96.9±3.1	96.9±3.1	97.3±3.1	97.2±3.1
	1NN	SVM-RFE	94.8±4.7	96.0±3.6	96.2±3.5	96.4±3.5	96.5±3.4
		En SVM-RFE	96.1±3.9	96.3±3.9	96.5±3.4	96.6±3.5	96.6±3.4
		IW SVM-RFE	96.0±4.6	95.9±4.5	95.8±4.4	95.8±4.3	95.9±4.3

7.3.2.3 Overall Conclusions

From the classification results based on both experiment settings with different evaluation measures, we can clearly observe that instance weighting can maintain even increase the predictive performance. Overall, the results proves that the improvement of feature selection stability can be achieved by instance weighting without sacrificing the classification accuracy.

Table 7.5. Classification performance measured by the AUC (average value ± standard deviation) of the linear SVM and 1NN for the Conventional, Ensemble (**En**), and Instance Weighting versions (**IW**) of ReliefF at increasing gene signature sizes for four data sets.

Data Set	Classifier	Selection Method	Number of Selected Features				
			10	50	100	150	200
Colon	SVM	ReliefF	78.8±8.8	80.1±8.8	78.5±8.7	77.5±8.9	76.1±8.5
		En ReliefF	78.9±8.9	80.2±9.9	79.1±9.4	77.3±9.6	76.1±9.0
		IW ReliefF	78.3±8.2	77.6±9.4	78.1±9.4	76.4±10.0	75.4±10.0
	1NN	ReliefF	73.0±8.4	73.4±8.2	75.4±8.3	76.5±7.7	77.0±8.8
		En ReliefF	72.1±9.7	73.9±8.9	74.0±8.8	75.5±8.5	75.3±9.6
		IW ReliefF	72.8±8.5	72.0±9.1	74.6±10.7	75.8±10.5	76.0±9.9
Leukemia	SVM	ReliefF	91.5±5.3	95.2±4.7	95.9±4.1	96.1±3.9	96.4±3.4
		En ReliefF	91.3±5.5	94.7±4.3	95.7±4.0	96.3±3.7	96.2±3.8
		IW ReliefF	91.2±5.6	94.5±5.2	95.7±4.7	95.2±4.9	95.3±5.0
	1NN	ReliefF	89.7±5.2	90.2±6.1	90.3±5.6	91.1±6.0	91.0±6.1
		En ReliefF	89.6±5.2	90.4±6.2	90.5±6.1	90.6±6.1	90.7±5.9
		IW ReliefF	90.3±5.8	89.7±7.2	91.5±6.7	89.8±7.0	90.2±7.9
Prostate	SVM	ReliefF	93.3±3.8	93.0±4.1	91.4±4.4	91.4±4.2	91.7±4.2
		En ReliefF	93.4±3.5	92.4±4.0	91.4±4.1	91.0±4.4	91.9±4.2
		IW ReliefF	93.3±3.8	92.7±3.8	91.4±4.1	91.3±4.7	91.4±4.1
	1NN	ReliefF	89.3±4.5	87.2±5.6	88.5±5.3	89.2±5.1	89.5±4.7
		En ReliefF	88.6±4.8	87.2±5.6	88.1±5.5	88.3±5.6	88.4±5.1
		IW ReliefF	89.3±4.7	87.9±5.4	89.2±5.4	89.3±5.1	89.5±5.1
Lung	SVM	ReliefF	96.2±4.2	96.7±3.2	96.7±3.3	97.0±3.3	97.4±3.0
		En ReliefF	97.0±3.0	97.0±3.1	97.1±3.1	97.2±3.2	97.5±3.0
		IW ReliefF	96.8±4.7	96.7±4.0	98.6±2.4	98.4±2.4	98.8±2.2
	1NN	ReliefF	96.0±4.5	95.7±4.4	95.4±4.0	95.2±4.2	94.9±4.3
		En ReliefF	93.0±8.9	96.8±3.3	96.2±3.6	95.8±3.8	95.6±4.0
		IW ReliefF	92.9±6.5	96.7±3.4	96.5±3.5	96.8±3.6	96.3±4.4

7.4 Consensus Gene Signatures

We further evaluate the stability of the three versions of the SVM-RFE method by examining the selection frequency of each gene across the 100 random training/test splits of a given dataset. Given a dataset, a selection method, and a gene signature size (e.g., 50) for each execution of the method, some genes are more consistently represented across the 100 generated gene signatures than others. A consensus gene signature can be constructed by extracting those genes frequently selected over many samplings. From a majority voting perspective, a gene is retained in the consensus

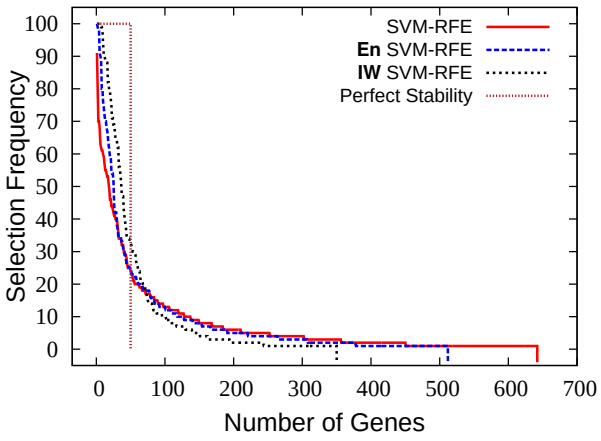

Figure 7.7. Selection frequency plots of the SVM-RFE, Ensemble (**En**) SVM-RFE and Instance Weighting (**IW**) SVM-RFE methods for the Colon data. Each plot shows how many genes occur in at least how many of the 100 gene signatures of size 50 selected by each method. The area under the curve of each method equals the area under the perfect stability curve (100×50). The more overlap between the two areas, the more stable the method is.

gene signature (hence called a consensus gene), if it is represented in more than 50% of all gene signatures generated by the same method. The threshold 50% may be increased to shorten the consensus signatures while increasing the confidence on the consensus genes.

Figure 7.7 shows the selection frequency curves of the three versions of the SVM-RFE methods for the Colon dataset. Each curve shows the selection frequencies of all features selected by a corresponding method at signature size 50 (features occurring in none of the 100 signatures not shown). As a reference, the frequency curve for perfect stability (i.e, the same 50 genes appearing in all of the 100 signatures) is also shown. The area under the curve of each selection method and the area under the perfect stability curve are equal to the size of the 100×50 signature matrix. The more overlap between the area under the curve of a selection method and the area under the perfect stability curve, the more stable the selection method is. Comparing the curves of the three methods, we can observe that the instance weighting method is more stable than the other two methods; its curve is closer to the perfect stability

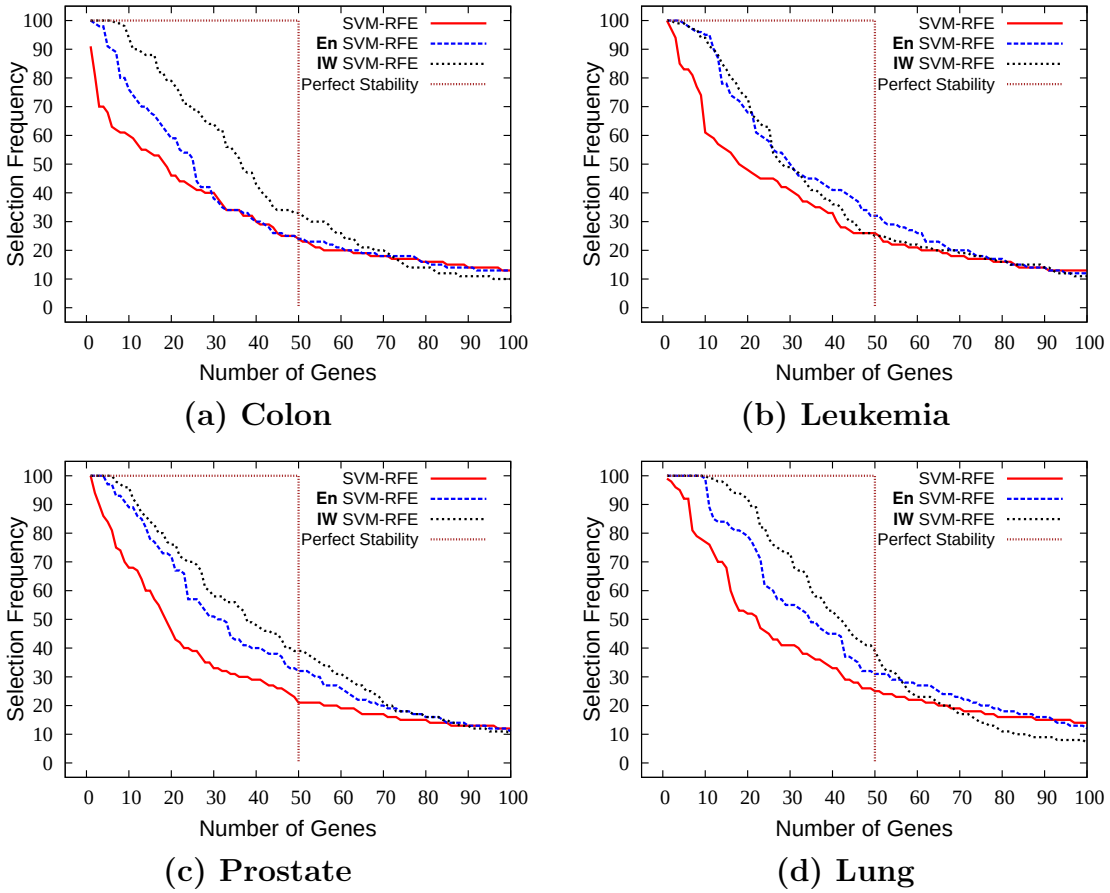

Figure 7.8. Selection frequency plots of the SVM-RFE, Ensemble (**En**) SVM-RFE and Instance Weighting (**IW**) SVM-RFE methods for Four Datasets. Each plot shows how many genes occur in at least how many of the 100 gene signatures of size 50 selected by each method. Only the top 100 most frequently selected genes are included. Fig. **(a) Colon** provides a "zoomed in" view of Fig. 7.7.

curve (on the left side of the perfect stability curve) and has a much shorter tail (on the right size of the perfect stability curve).

Figure 7.8 (a) provides a "zoomed in" view of the frequency plots in Figure 7.7 by focusing on the top 100 most frequently selected genes for the Colon dataset. Clearly, the instance weighting method consistently selects more consensus genes at various frequency threshold levels from 50% to 100% than the other two methods. For example, at the 50% threshold level, the SVM-RFE, ensemble, and instance weighting methods respectively select 18, 25, and 36 consensus genes. If the threshold level is

Table 7.6. The numbers of genes above certain selection frequencies across 100 gene signatures of size 50 selected by the SVM-RFE, Ensemble (**En**) SVM-RFE and Instance Weighting (**IW**) SVM-RFE.

		Frequency Intervals		
Data	Selection Method	[1,100]	(50,100]	(85,100]
Colon	SVM-RFE	642	18	1
	En SVM-RFE	512	25	7
	IW SVM-RFE	**350**	**36**	**16**
Leukemia	SVM-RFE	688	18	4
	En SVM-RFE	**397**	**30**	13
	IW SVM-RFE	469	28	**14**
Prostate	SVM-RFE	722	18	4
	En SVM-RFE	371	32	13
	IW SVM-RFE	**262**	**37**	**15**
Lung	SVM-RFE	558	22	6
	En SVM-RFE	308	34	12
	IW SVM-RFE	**246**	**42**	**22**

increased to 85%, the consensus gene signature sizes for the three methods will shrink to 1, 7, and 16, respectively. These numbers are also reported in Table 7.6.

We performed the same analysis as above for all the four datasets used in this study. For the sake of conciseness of presentation, figures showing the full view of the frequency plots (as Figure 7.7) for the Leukemia, Prostate, and Lung datasets are not included. The "zoomed in" view of the frequency plots for each dataset is provided in Figure 7.8. Furthermore, Table 7.6 precisely reports the total numbers of genes that are selected in at least one of the 100 generated gene signatures as well as the numbers of consensus genes with frequency over 50 and 85 for each method on each microarray dataset.

From Figure 7.8 and Table 7.6, we can observe that for all the four datasets used in our study, the instance weighting method significantly improves the stability of the SVM-RFE method. Comparing the instance weighting method with the ensemble method, the former clearly outperforms the latter for the Colon, Prostate, and Lung datasets. For the Leukemia dataset, the two methods perform very similar, with the ensemble method being slightly better. Such observations are consistent with those from in Section 7.3, where the stability performance is measured with respective to

pairwise gene signature similarity at various signature sizes. Figure 7.8 and Table 7.6 in this section provide a more detailed view about the stability of the three versions of the SVM-RFE methods at signature size 50 for each dataset as shown in Figure 7.5 and Figure 7.6. The analysis on consensus gene signatures also demonstrates the impact of the stability improvement by the instance weighing method. The instance weighting method enables the SVM-RFE method to consistently select much fewer genes of low frequency and produce much bigger consensus gene signatures, which in turn improves the interpretability and reliability of the generated gene signatures.

7.5 Algorithm Efficiency

Observations from this chapter indicate that different feature selection algorithms can lead to similarly good classification results, while their stability performance can largely vary. The difficulty in distinguishing feature selection algorithms in terms of predictive accuracy mainly lies in the small sample size of the test sets in microarray data as opposed to synthetic data used in Chapter 5. Studying the stability of feature selection provides a new perspective to domain experts in choosing a feature selection algorithm and validating the selected features.

Figure 7.9 compares the running time of the three versions of SVM-RFE and ReliefF on the entire data set for each microarray data set. **En** SVM-RFE is almost 20 times slower than SVM-RFE, which is caused by the repeatedly process of feature selection on every bootstrapped training data, while **IW** SVM-RFE is only slightly slower than SVM-RFE (only a little overhead for learning the weight for each instance). The same trend applies to the three versions of ReliefF. The efficiency of **IW** SVM-RFE and **IW** ReliefF lies in the fact that the instance weighting process acts as a preprocessing step which is executed only once. Such slight extra cost of instance weighting leads to significantly increased stability for the base algorithms.

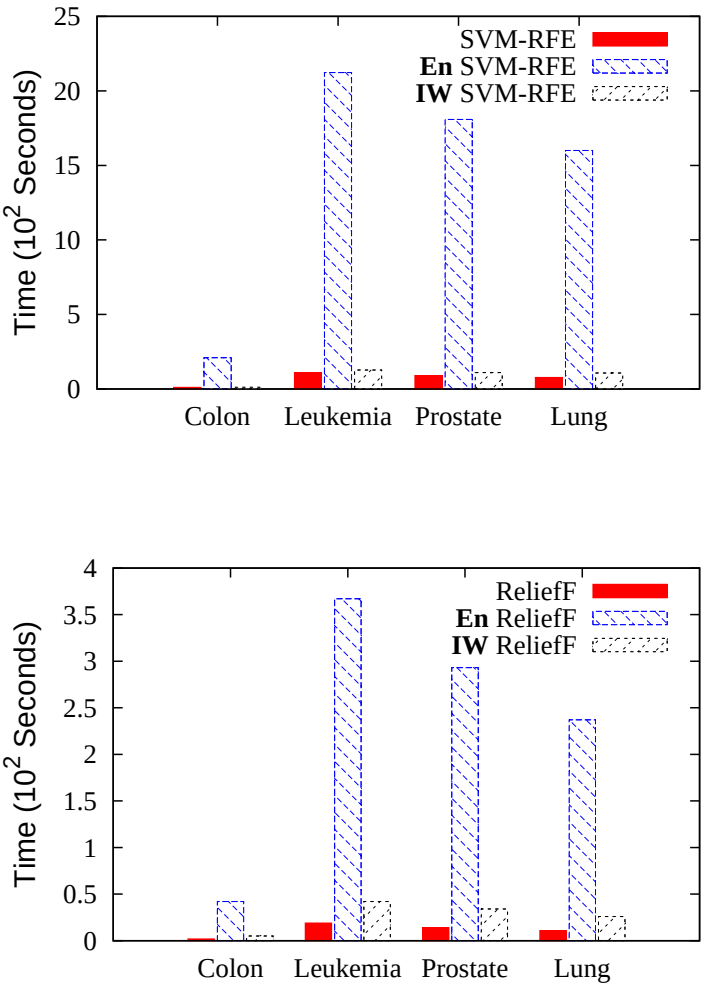

Figure 7.9. Running time for the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of the SVM-RFE and ReliefF algorithms on microarray data.

8 Conclusion and Future Work

Feature selection has been playing an important role in knowledge discovery from many application domains, and a great variety of feature selection algorithms are proposed and studied. In this dissertation, we focused on the stability of feature selection and contributed to the study of stable feature selection through theoretical analysis, proposal of novel algorithms and also extensive empirical evaluation. Since this is a new research direction in feature selection with very few existing studies, there is plenty of room for future study. In this chapter, we summarize our work in this dissertation and identify some future research directions.

8.1 Conclusion

The stability of feature selection is an interesting and important issue. The development of stable feature selection algorithms is drawing increasing attention from researchers in various domains. Although the study presented in this dissertation is not the first to raise the stability issue of feature selection, it has made an initial effort on providing a theoretical framework and a principled approach for developing stable feature selection algorithms.

In this dissertation, we have presented a theoretical framework about feature selection stability based on a formal bias-variance decomposition of feature selection error. The idea of decomposing feature selection error into bias and variance is illuminated by a similar decomposition of classification error under which the stability of learning algorithms are formally defined and studied. With the help of this theoretical

framework, we have revealed the relationship between the stability and the accuracy of feature selection. We have also demonstrated that one does not have to sacrifice predictive accuracy in order to get more stable feature selection results and better tradeoff between the bias and the variance of feature selection can lead to more stable results while maintaining or even improving predictive accuracy based on the selected features. Moreover, we have uncovered the dependency of feature selection stability on the number of instances in a training set (or sample size), and suggested a variance reduction approach for improving the stability of feature selection algorithms under small sample size.

After a thorough study on the stability of feature selection from the theoretical perspective, we have proposed an empirical variance reduction framework - margin based instance weighting. The framework first weights each instance in a training set according to its importance to feature evaluation, and then provides the weighted training set to a feature selection algorithm. There are two challenges under this empirical framework. The first is to learn weight for each instance from the aspect of improving feature selection stability. To this end, we have introduced a novel concept, margin vector feature space, which enables the the estimation of the importance of instances with respect to (w.r.t.) feature evaluation. We differentiated this new feature space from the original space with very concrete example. We also proposed an instance weighting approach based on the transformed space which assign higher weights to instances from regions which contribute more to the aggregate feature weights and assign lower weights to instances from other less important (or outlying) regions. The second challenge is how to present the weighted instances to existing feature selection algorithms. Based on theoretical analysis and empirical evaluation, we have seamlessly incorporated weights into two state-of-the-art approaches, SVM-RFE and ReliefF under study.

The proposed theoretical and empirical frameworks have been validated through an extensive set of experiments. We have adopted various stability metrics and classification performance measures and also different experimental settings (both cross-validation procedure and bootstrapping procedure) to achieve the comprehensive evaluation of the experimental results. We have also carefully chosen the feature selection algorithms and classification algorithms to better serve the need of researching in feature selection community.

Experiments on synthetic data sets have demonstrated the bias-variance decomposition of feature selection error and the effects of sample size on feature selection, using the SVM-RFE algorithm. These experiments have also verified the effectiveness of the proposed instance weighting framework at reducing the variance of feature weighting by SVM-RFE, and in turn improving the stability and the predictive accuracy of the selected features by SVM-RFE.

Experiments on real-world microarray data sets have further verified that the instance weighting framework is effective at reducing the variance of feature weighting and improving the subset stability for two representative feature selection algorithms, SVM-RFE and ReliefF, while maintaining comparable predictive accuracy based on the selected features. Moreover, the instance weighting framework has been shown to be more effective and efficient than a recently proposed ensemble framework for stable feature selection.

8.2 Future Work

The study in this dissertation opens a number of directions for future research, such as: alternative instance weighting approach, extension to other feature selection algorithms, study on how bias-variance properties of feature selection affect the classification accuracy of learning models built on the selected features and study on various factors for stability of feature selection.

8.2.1 Extension to Other Feature Selection Algorithms

We have studied how to apply the instance weighting framework to two of the state-of-the-art feature selection algorithms, SVM-RFE and ReliefF, and have evaluated its effectiveness of improving the stability of those two baseline algorithms, and have also showed its superiors performance over another recently proposed scheme for stable feature selection, the ensemble approach. But our weighting approach can be extended to more feature selection algorithms beyond SVM-RFE and ReliefF.

Intuitively, the instance weighting framework doesn't have any restriction on the choice of baseline feature selection methods as long as they have the capability of handling weighted instances. And thus the key issue behind the generalization of instance weighting approach is how to present the weighted instances to the conventional feature selection algorithms appropriately. As discussed in our previous study, instance weight affect the trade-off between sample margin and error penalty when deciding the optimistic hyperplane, which is used for assigning feature weight in SVM-RFE. Similarly, instance weight could have more influence on the evaluation of feature importance for different feature selection algorithms. It is an important issue of studying how to appropriately incorporate instance weights to each feature selection algorithm.

8.2.2 Alternative Instance Weighting Schemes

Although the sample weight learned from the Margin Vector Feature Space is proved both effective and efficient for improving the stability of conventional feature selection algorithms, margin is not the only way for evaluating weight of samples w.r.t their influence on the selection of features. Under the general framework of instance weighting, we can seek the best weighting scheme for tailoring specific feature selection algorithm instead of the margin-based instance weighting method with general purpose as proposed in this dissertation.

Another issue with the current margin-based instance weighting framework comes from the weighting metric used for assigning weight for each sample after the Hypothesis-margin Feature Space transformation. Although it's reasonable to measure the Euclidean distance among the pairwised instances in the transformed space indicating instance importance, the weighting metrics itself seems heuristic, which could raise doubts on the real representation of instance importance from the Margin Vector Feature Space, especially for high-dimensional data. In fact, different weighting metrics can be applied and verified as the study on stability of feature selection is further explored. The relationship between the weighting metrics and its influence on the feature selection results should also be studied, which will appropriately guide the choice of weighting metrics and achieve even better stability results eventually.

8.2.3 Study on How Bias-Variance Properties of Feature Selection Affect Classification Accuracy

Under the theoretical framework of bias-variance decomposition of feature selection error, we proposed an empirical variance reduction framework which improves the stability of feature selection without sacrificing the prediction accuracy. But more specifically, how stability of feature selection will further affect the bias-variance properties of learning models has never been studied before. Since most domain experts still focus on the classification performance based on the selected features while evaluating the usefulness of proposed feature selection algorithms, the relationship between feature selection stability and prediction accuracy is definitely a critical issue worthwhile to be explored in the future.

8.2.4 Study on Various Factors for Stability of Feature Selection

The stability of feature selection is a complicated issue. Besides sample size, recent studies on this issue [Kalousis et al. 2007; Loscalzo et al. 2009] have shown that the

stability of feature selection results depends on various factors such as data distribution, mechanism of feature selection and so on. The trade-off between relevance and redundance of features during the selection process will also impact the stability of feature selection methods. It would be interesting to have an in-depth understanding of those issues and identify why and how those different factors influence the feature selection stability. And it will illuminate the development of stable feature selection algorithms in the future.

Bibliography

Agrawal, R., Imielinski, T., and Swami, A. 1993. Database mining: A performance perspective. *IEEE Trans. on Knowledge and Data Engineering 5,* 6 (Dec.), 914–925.

Aizerman, M., Braverman, E., and Rozonoer, L. 1964. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control 25*, 821–837.

Alon, U., Barkai, N., Notterman, D. A., Gishdagger, K., Ybarradagger, S., Mackdagger, D., and Levine, A. J. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl Acad. Sci. USA 96*, 6745–6750.

Appice, A., Ceci, M., Rawles, S., and Flach, P. 2004. Redundant feature elimination for multi-class problems. In *Proceedings of the 21st International Conference on Machine learning.* 33–40.

Au, W., Chan, K. C. C., Wong, A. K. C., and Wang, Y. 2005. Attribute clustering for grouping, selection, and classification of gene expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics 2,* 2, 83–101.

Bartlett, P. and J., S.-T. 1999. Generalization performance of support vector machines and other pattern classifiers. *Advances in Kernel Methods – Support Vector Learning*, 43–54.

Belkin, M. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation 15*, 1373–1396.

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

Blum, A. L. and Rivest, R. L. 1992. Training a 3-node neural networks is NP-complete. *Neural Networks 5*, 117 – 127.

Bradley, P. S. and Mangasarian, O. L. 1998. Feature selection via concave minimization and support vector machines. In *Proceedings of Fifteenth International Conference on Machine*

Learning. 82–90.

Butterworth, R., Piatetsky-Shapiro, G., and Simovici, D. A. 2005. On feature selection through clustering. In *Proceedings of the Fifth IEEE International Conference on Data Mining*. 581–584.

Caruana, R. and Freitag, D. 1994. Greedy attribute selection. In *Proceedings of the Eleventh International Conference on Machine Learning*. 28–36.

Child, D., Ed. 2006. *Essentials of Factor Analysis*. Continuum.

Clarke, R., Ressom, H., A., W., J., X., and etc. 2008. The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature Reviews Cancer 8*, 37–49.

Cortes, C. and Vapnik, V. 1995. Support vector networks. *Machine Learning 20*, 273–297.

Crammer, K., Gilad-Bachrach, R., and Navot, A. 2002. Margin analysis of the LVQ algorithm. In *Proceedings of the 17th Conference on Neural Information Processing Systems*. 462–469.

Dash, M., Choi, K., Scheuermann, P., and Liu, H. 2002. Feature selection for clustering – a filter solution. In *Proceedings of the Second International Conference on Data Mining*. 115–122.

Dash, M. and Liu, H. 1997. Feature selection for classification. *Intelligent Data Analysis: An International Journal 1,* 3, 131–156.

Davis, C. A., Gerick, F., Hintermair, V., Friedel, C. C., Fundel, K., Kffner, R., and Zimmer, R. 2006. Reliable gene signatures for microarray classification: assessment of stability and performance. *Bioinformatics 22*, 2356–2363.

Devijver, P. and Kittler, J. 1982. *Pattern Recognition: A Statistical Approach*. Prentice Hall International.

Ding, C. and Peng, H. 2003. Minimum redundancy feature selection from microarray gene expression data. In *Proceedings of the Computational Systems Bioinformatics conference (CSB'03)*. 523–529.

Domingos, P. 2000. A unified bias-variance decomposition and its applications. In *Proceedings of the Seventeenth Internationl Conference on Machine Learning*. 231–238.

Donoho, D. and Grimes, C. 2003. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences of the United States*

of America 100, 10, 5591–5596.

Dy, J. G. and Brodley, C. E. 2000. Feature subset selection and order identification for unsupervised learning. In *Proceedings of the Seventeenth International Conference on Machine Learning*. 247–254.

Forman, G. 2003. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research 3*, 1289–1305.

Freund, Y. and Schapire, R. 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Computer Systems and Science 55,* 1, 119 – 139.

Fukunaga, K. 1990. *Introduction to Statistical Pattern Recognition*, 2 ed. San Diego: Academic Press.

Geman, S., Bienenstock, E., and Doursat, R. 1992. Neural networks and the bias/variance dilemma. *Neural Computation 4,* 1, 1–58.

Gilad-Bachrach, R., Navot, A., and Tishby, N. 2004. Margin based feature selection: theory and algorithms. In *Proceedings of the 21st International Conference on Machine learning*.

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

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

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

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

Hastie, T., Rosset, S., Tibshirani, R., and Zhu, J. 2004. The entire regularization path for the support vector machine. *The Journal of Machine Learning Research 5*, 1391–1415.

Hastie, T., Tibshirani, R., and Friedman, J. 2001. *The Elements of Statistical Learning*. Springer.

John, G. H., Kohavi, R., and Pfleger, K. 1994. Irrelevant feature and the subset selection problem. In *Proceedings of the Eleventh International Conference on Machine Learning*. 121–129.

Jornsten, R. and Yu, B. 2003. Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics 19*, 1100–1109.

Kalousis, A., Prados, J., and Hilario, M. 2007. Stability of feature selection algorithms: a study on high-dimensional spaces. *Knowledge and Information Systems 12*, 95–116.

Kewley, R. H., Embrechts, M. J., and Breneman, C. M. 2000. Data strip mining for the virtual design of pharmaceuticals with neural networks. *IEEE Transactions on Neural Networks 11,* 3, 668–679.

Kim, Y., Street, W., and Menczer, F. 2000. Feature selection for unsupervised learning via evolutionary search. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 365–369.

Kohavi, R. and John, G. H. 1997. Wrappers for feature subset selection. *Artificial Intelligence 97,* 1-2, 273–324.

Koller, D. and Sahami, M. 1996. Toward optimal feature selection. In *Proceedings of the Thirteenth International Conference on Machine Learning*. 284–292.

Kuncheva, L. 2007. A stability index for feature selection. In *Proceedings of the 25th International Multi-Conference on Artificial Intelligence and Applications*. 390–395.

Li, T., Zhang, C., and Ogihara, M. 2004. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics 20*, 2429–2437.

Liu, H. and Motoda, H., Eds. 2001. *Instance Selection and Construction for Data Mining*. Boston: Kluwer Academic Publishers.

Liu, H. and Setiono, R. 1996. Feature selection and classification - a probabilistic wrapper approach. In *Proceedings of the Ninth International Conference on Industrial and Engineering Applications of AI and ES*, T. Tanaka, S. Ohsuga, and M. Ali, Eds. Fukuoka, Japan, 419–424.

Liu, H. and Yu, L. 2005. Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on Knowledge and Data Engineering (TKDE) 17,* 4, 491–502.

Loscalzo, S., Yu, L., and Ding, C. 2009. Consensus group based stable feature selection. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and*

Data Mining (KDD-09). 567–576.

Maaten, L., Postma, E., and Herik, J. 2009. Dimensionality reduction: A comparative review. *Tilburg centre for Creative Computing, Tilburg University*.

Mitchell, T. M., Hutchinson, R., Niculescu, R. S., Pereira, F., Wang, X., Just, M., and Newman, S. 2004. Learning to decode cognitive states from brain images. *Machine Learning 57,* 1-2, 145–175.

Pepe, M. S., Etzioni, R., Feng, Z., Potter, J. D., Thompson, M. L., Thornquist, M., Winget, M., and Yasui, Y. 2001. Phases of biomarker development for early detection of cancer. *J Natl Cancer Inst 93*, 1054–1060.

Perkins, S., Lacker, K., and Theiler, J. 2003. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research 3*, 1333–1356.

Pinkel, D., Segraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., Collins, C., Kuo, W., Chen, C., Zhai, Y., Dairkee, S. H., Ljung, B., Gray, J. W., and Albertson, D. G. 1998. High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays. *Nature Genetics 20*, 207–211.

Pyle, D. 1999. *Data Preparation for Data Mining*. Morgan Kaufmann Publishers.

Robnik-Sikonja, M. and Kononenko, I. 2003. Theoretical and empirical analysis of Relief and ReliefF. *Machine Learning 53*, 23–69.

Roweis, S. and Saul, L. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science Magazine 290*, 2323–2326.

Rubinstein, B. Y. 1981. *Simulation and the Monte Carlo Method*. John Wiley & Sons, New York.

Saeys, Y., Abeel, T., and Peer, Y. V. 2008. Robust feature selection using ensemble feature selection techniques. In *Proceedings of the ECML Confernce*. 313–325.

Sebastiani, F. 2005. Text categorization. *Text mining and its applications to intelligence, CRM and Knowledge Management*.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., Lander, E. S., Loda, M., Kantoff, P. W., Golub, T. R., and Sellers, W. R. 2002. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell. 2*, 203–209.

Sun, Y. and Li, J. 2006. Iterative RELIEF for feature weighting. In *Proceedings of the 23rd international conference on Machine learning*. 913–920.

Turney, P. 1995. Technical note: bias and the quantification of stability. *Machine Learning 20,* 1-2, 23–33.

Wang, L., Zhu, J., and Zou, H. 2007. Hybrid huberized support vector machines for microarray classification. In *Proceedings of the 24th International Conference on Machine learning*. 983 – 990.

Wang, L. P., Chu, F., and Xie, W. 2007. Accurate cancer classification using expressions of very few genes. *IEEE Transactions on Bioinformatics and Computational Biology 4,* 1, 40–53.

Wasikowski, M. and Chen, X. 2010. Combating the small sample class imbalance problem using feature selection. *IEEE Transactions on Knowledge and Data Engineering 22,* 10, 1388–1400.

Weinberger, K. and Saul, L. 2004. Unsupervised learning of image manifolds by semidefinite programming. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Witten, I. H. and Frank, E. 2005. *Data Mining - Pracitcal Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers.

Yu, L., Ding, C., and Loscalzo, S. 2008. Stable feature selection via dense feature groups. In *Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (KDD-08)*. 803–811.

Yu, L. and Liu, H. 2003. Feature selection for high-dimensional data: a fast correlation-based filter solution. In *Proceedings of the twentieth International Conference on Machine Learning (ICML)*. 856–863.

Yu, L. and Liu, H. 2004. Redundancy based feature selection for microarray data. In *Proceedings of the Tenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 737–742. http://portal.acm.org/citation.cfm?id=1014052.1014149.

Zhang, M., Zhang, L., Zou, J., Yao, C., Xiao, H., Liu, Q., Wang, J., Wang, D., Wang, C., and Guo, Z. 2009. Evaluating reproducibility of differential expression discoveries in microarray studies by considering correlated molecular changes. *Bioinformatics 25,* 13, 1662–1668.