2010 IEEE International Conference on Data Mining

A Variance Reduction Framework for Stable Feature Selection

Yue Han, Lei Yu
Department of Computer Science
Binghamton University
Binghamton, NY 13902-6000, USA
yhan1@binghamton.edu, lyu@cs.binghamton.edu

***Abstract*—Besides high accuracy, stability of feature selection has recently attracted strong interest in knowledge discovery from high-dimensional data. In this study, we present a theoretical framework about the relationship between the stability and accuracy of feature selection based on a formal bias-variance decomposition of feature selection error. The framework also suggests a variance reduction approach for improving the stability of feature selection algorithms. Furthermore, we propose an empirical variance reduction framework, margin based instance weighting, which weights training instances according to their influence to the estimation of feature relevance. We also develop an efficient algorithm under this framework. Experiments based on synthetic data and real-world microarray data verify both the theoretical framework and the effectiveness of the proposed algorithm on variance reduction. The proposed algorithm is also shown to be effective at improving subset stability, while maintaining comparable classification accuracy based on selected features.**

***Keywords*-feature selection; stability; bias-variance decomposition; variance reduction; high-dimensional data**

I. Introduction

Various feature selection algorithms have been developed with a focus on improving classification accuracy while reducing dimensionality [1], [2], [3]. 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 characteristic markers to explain the observed phenomena. For example, in microarray analysis, biologists are interested in finding a small number of features (genes or proteins) that explain the mechanisms driving different behaviors of microarray samples [4]. A feature selection algorithm often selects largely different subsets of features under variations to the training data, although most of these subsets are as good as each other in terms of classification performance [5], [6], [7]. Such instability dampens the confidence of domain experts in experimentally validating the selected features.

The stability of feature selection is a complicated issue. Recent studies on this issue [6], [7] have shown that the stability of feature selection results depends on various factors such as data distribution, mechanism of feature selection, and sample size. Moreover, the stability of feature selection results should be investigated together with the predictive performance of 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 instances) that yields very stable feature subsets but bad predictive models.

In this study, we present a theoretical framework about feature selection stability based on a formal bias-variance decomposition of feature selection error. The theoretical framework explains the relationship between the stability and 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 variance of feature selection can lead to more stable results while maintaining or even improving predictive accuracy based on the selected features.

Furthermore, we propose an empirical framework, variance reduction via margin based instance weighting, to achieve such a better tradeoff. The main idea of this framework is to first weight each instance in a training set according to its influence to the estimation of feature relevance, and then provide the weighted training set to a feature selection algorithm. Intuitively, different instances in a training set could have different influence on the feature selection result according to their views (or local profiles) of the relevance of each feature. If an instance shows a noticeably distinct local profile from the other instances, its absence or presence in the training data will substantially affect the feature selection result. In order to reduce the variance of feature selection result, instances with outlying local profiles need to be weighted differently from the rest of the instances. To this end, we develop an efficient margin based instance weighting algorithm which assigns a weight to each instance according to the outlying degree of its local profile of feature relevance compared with other instances. The local profile of feature relevance at a given instance is measured based on the hypothesis margin of the instance.

Our experiments on synthetic data demonstrate the bias-variance decomposition of feature selection error based on the widely adopted SVM-RFE algorithm. These experiments also verify the effectiveness of the proposed instance weighting algorithm at reducing the variance of feature weighting

1550-4786/10 $26.00 © 2010 IEEE
DOI 10.1109/ICDM.2010.144

by SVM-RFE, and in turn improving the stability and predictive accuracy of the selected features by SVM-RFE. Experiments on a set of public microarray data sets further verify that the instance weighting algorithm is effective at reducing the variance of feature weighting, improving the stability of the selected subsets, while maintaining comparable predictive accuracy based on the selected features by SVM-RFE. Moreover, the instance weighting algorithm is shown to be more effective and efficient than a recently proposed ensemble feature selection method.

The rest of the paper is organized as follows. Section II reviews related work in contrast with our work. Section III introduces our theoretical framework on stability of feature selection. Section IV describes an empirical framework of margin based instance weighting and an efficient algorithm developed under this framework. Section V evaluates the theoretical and empirical frameworks based on synthetic and microarray data. Section VI concludes the paper and outlines future research directions.

II. Related Work

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 [6], [8]. 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 [9] which aggregates the results from a conventional feature selection algorithm repeatedly applied on a number of bootstrapped samples of the same training set. Loscalzo *et al*. proposed an alternative approach which exploits the intrinsic correlations among a large number of features to identify consensus feature groups and then selects relevant feature groups [7]. In contrast to existing studies on stable feature selection, our study provides a theoretical framework which explains the relationship between the stability and accuracy of feature selection. In addition, our study proposes an instance weighting framework for improving the stability of feature selection algorithms.

Another line of closely related research is margin based feature selection. Several studies have developed feature selection algorithms under the large margin principles, such as SVM-based feature selection [10] and the Relief family of algorithms [11], [12], [13]. These studies have shown both nice theoretical properties and good generalization performance of margin based feature selection algorithms, but have not yet addressed the stability issue of feature selection. 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 [13]) 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.

A problem related to the stability of feature selection is the stability of learning algorithms. It is well known that the generalization error of a learning algorithm can be decomposed into bias, variance, and noise. Previous studies on the bias-variance tradeoff [14], [15], [16] explain the relationship between the stability and accuracy of learning algorithms, while our study reveals the relationship between the stability and accuracy of feature selection algorithms.

III. Theoretical Framework

In this section, 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 accuracy of feature selection based on this decomposition.

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 paper, 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 can 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. $r(D)$ can be viewed as a Monte Carlo estimate of r^* for feature weighting algorithms which decide the relevance score of each feature based on aggregating the scores over all instances in a training set (e.g., the Relief family of algorithms and SVM-based algorithms). To evaluate the overall performance of $\mathcal{F}$, the quantity of interest is the *expected loss (or error)*, $EL(X)$, defined as:

$$EL(X) = \mathbb{E}[(r^* - r(D))^2] = \sum_{D \in \mathscr{D}} (r^* - r(D))^2 p(D) \ , \quad (1)$$

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:

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

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

$$Var(X) = \mathbb{E}[r(D) - \mathbb{E}(r(D))]^2 = \sum_{D \in \mathscr{D}} [r(D) - \mathbb{E}(r(D))]^2 p(D) \ . \quad (3)$$

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:

$$\frac{1}{d}\sum_{j=1}^{d} EL(X_j) = \frac{1}{d}\sum_{j=1}^{d} Bias(X_j) + \frac{1}{d}\sum_{j=1}^{d} Var(X_j) \ . \quad (4)$$

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

The above bias-variance decomposition reveals the relationship between the stability (the opposite of variance) and the accuracy (the opposite of error) of feature selection. Reducing either the bias or the variance alone does not necessarily reduce the error, but a better tradeoff between the bias and the variance does. One thing to note at this point is that the error 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 [17]. However, in practice, the generalization error depends on both the error of feature selection and the bias-variance properties of the learning algorithm itself.

The framework presented here also sheds lights on the relationship between the stability of feature selection and the predictive accuracy based on the selected features. Existing studies on stable feature selection [6], [9] 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 propose an empirical framework to achieve such a better tradeoff for feature weighting algorithms.

IV. Empirical Framework: Margin Based Instance Weighting

The empirical framework is motivated by importance sampling, one of the commonly used variance reduction techniques [18]. The theory of importance sampling suggests that in order to reduce the variance of a Monte Carlo estimator (e.g., the estimate of feature relevance by a feature weighting algorithm based on a training set), 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. Section IV-A presents the main ideas of the proposed framework of instance weighting for variance reduction. Section IV-B provides the technical details of the margin based instance weighting algorithm developed under this framework.

A. Margin Vector Feature Space

Margins [19] 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 an instance w.r.t. a hypothesis [20]. Sample margin as used by SVMs [19] measures the distance between the instance and the decision boundary of the hypothesis. Hypothesis margin as used by AdaBoost [21] measures the distance between the hypothesis and the closest hypothesis that assigns an alternative label to the given instance. Feature selection

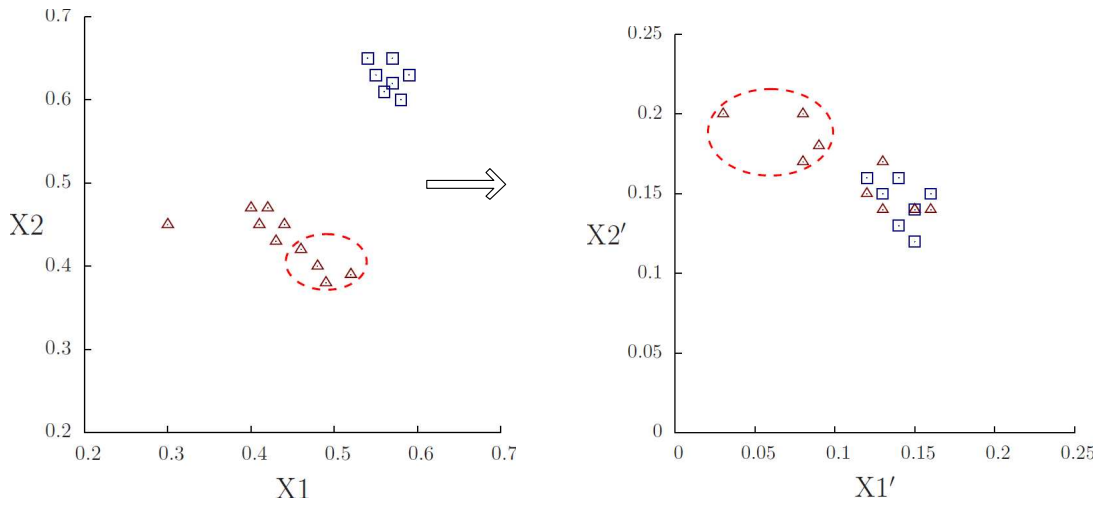

Figure 1. An illustrative example for Margin Vector Feature Space. Each data point in the original feature space (left) is projected to the margin vector feature space (right) according to its hypothesis margin in the original feature space.

algorithms developed under the large margin principles [10], [12] evaluate the relevance of features according to their respective contributions to the margins.

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 1: Let $\mathbf{x} = (x_1, ..., x_d)$ be an instance in the original feature 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 $\Re'^d$ according to

$$x'_j = |x_j - x_j^M| - |x_j - x_j^H| \ , \tag{5}$$

where x'_j is the jth coordinate of $\mathbf{x}'$ in the new feature space $\Re'^d$, 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 $\Re'^d$ 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. Figure 1 illustrates the idea 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. (5). We can clearly see that instances labeled with triangles exhibit largely different outlying degrees in the two feature spaces. Specifically, those in the dashed ovals are evenly distributed within the proximity to the rest of the triangles (except the outlier on the leftmost) in the original feature space, but are clearly separated from the majority of the instances in the margin vector feature space. The outlier triangle in the original space becomes part of the the majority group in the margin vector feature space. To decide the overall relevance of feature $X1$ vs. $X2$, one intuitive idea is to take the average over all margin vectors, as adopted by the well-known Relief algorithm [13]. However, since the triangles in the dashed oval exhibit distinct margin vectors from the rest of the instances, the presence or absence of these instances will affect the global decision on which feature is more relevant.

From this illustrative example, we can see that 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. By identifying and reducing the emphasis on these outlying instances, more stable results can be produced from a feature selection algorithm. In the next section, we will further discuss how to exploit such discrepancy to weight instances in order to alleviate the affect of training data variations on feature selection results.

B. Margin Based Instance Weighting Algorithm

The previous 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. (5) can then be extended to:

$$x'_j = \sum_{l=1}^{m} |x_j - x_j^{M_l}| - \sum_{l=1}^{h} |x_j - x_j^{H_l}| \ , \tag{6}$$

Algorithm 1 Margin Based Instance Weighting

Input: training data $D = \{\mathbf{x}_i\}_{i=1}^n$
Output: weight vector $\mathbf{w}$ for all instances in D
// 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. (6)
  end for
end for
// 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. (7)
end for

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).

Once the margin vector feature space is generated, the next task is to exploit the discrepancy of margin vectors in this space to weight instances in the original space. To quantitatively evaluate the outlying degree of each margin vector $\mathbf{x}'$, we measure the average distance of $\mathbf{x}'$ to all other margin vectors; greater average distance indicates higher outlying degree. As illustrated in Figure 1, the global decision of feature relevance is more sensitive to instances that largely deviate from the rest of the instances in the margin vector feature space than to instances that have low outlying degrees. To improve the stability of a feature selection algorithm under training data variations, we assign lower weights to instances with higher outlying degrees. This decision is consistent with the intuition behind importance sampling introduced earlier. Specifically, the weight for an instance $\mathbf{x}$ in the original feature space is given by the following formula:

$$w(\mathbf{x}) = \frac{1/\overline{dist}(\mathbf{x}')}{\sum_{i=1}^n 1/\overline{dist}(\mathbf{x}'_i)} \ , \tag{7}$$

where

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

Algorithm 1 outlines the key steps of margin based instance weighting. 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., $n << d$).

V. Empirical Study

The objective of empirical study is threefold: (1) to demonstrate the bias-variance decomposition proposed in Section III; (2) to verify the effectiveness of the proposed instance weighting algorithm on variance reduction; and (3) to verify the effect of variance reduction on improving the stability and predictive performance of the selected subsets. Section V-A introduces the subset stability measure used in our experiments. In Section V-B, 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 algorithm significantly reduces the variance of feature weights assigned by SVM-RFE, and consequently, improves both the stability and the classification accuracy of the selected feature subsets. In Section V-C, we further verify the effectiveness of the instance weighting algorithm on variance reduction and stability improvement based on real-world microarray data sets. Moreover, we show that the instance weighting algorithm is more effective and efficient than a recently proposed ensemble feature selection method.

A. Subset Stability Measure

The variance defined in Section III 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 [6], [7], 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

$$\overline{Sim} = \frac{2\sum_{i=1}^q \sum_{j=i+1}^q Sim(S_i, S_j)}{q(q-1)} \ , \tag{8}$$

where $Sim(S_i, S_j)$ represents a similarity measure between two subsets. For specific measure, we adopt the Kuncheva index, suggested by [8], defined as follows:

$$Sim(S_i, S_j) = \frac{|S_i \cap S_j| - (k^2/d)}{k - (k^2/d)} \ , \tag{9}$$

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 in the index 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.

B. Experiments on Synthetic Data

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. The first 5 groups of features are equally relevant, and the rest of the features are irrelevant.

To measure the variance, bias, and error of a given feature selection algorithm according to the definitions in Section III, 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 SVM-RFE [10], a widely adopted feature selection algorithm for high-dimensional data. The main process of SVM-RFE is to recursively eliminate features of low weights, using SVM to determine feature weights. 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 absolute values of feature weights in the optimal hyperplane, and eliminates one or more features with the lowest weights. This recursive feature elimination (RFE) process stops until all features have been removed or a desired number of features is reached. In our implementation, 10 percent of the remaining features are eliminated at each iteration (as suggested by the authors of the algorithm). We used Weka's implementation [22] of SVM (linear kernel, default C parameter).

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.

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. (1)-(4)) are then calculated at each iteration of the RFE process. 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 its variance, bias, and error under instance weighting. We refer to the instance weighting version of SVM-RFE as **IW** SVM-RFE.

Figure 2 reports the variance, bias, and error of SVM-RFE based on both the original and weighted training sets across the RFE process (until 10 features remain at the 40th iteration). We can observe the following three major trends. 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. (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.

3) Stability and Predictive Performance w.r.t. Selected Subsets: We next verify the effect of variance reduction

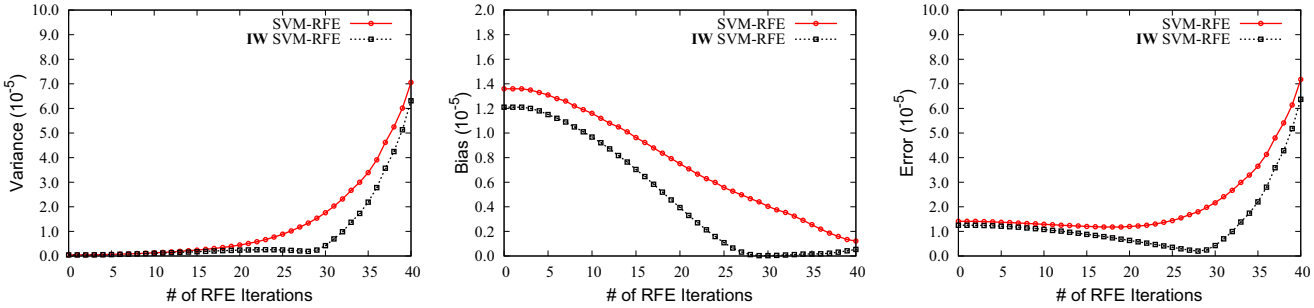

Figure 2. Variance, Bias, and Error of the feature weights assigned by the conventional and Instance Weighting (IW) versions of the SVM-RFE algorithm at each iteration of the Recursive Feature Elimination (RFE) process for synthetic data.

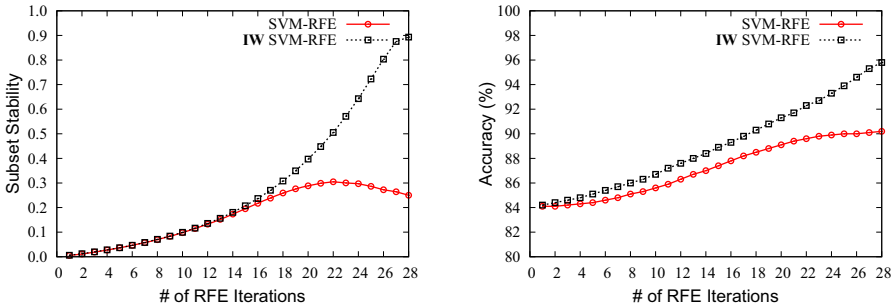

Figure 3. Stability (by Kuncheva Index) and predictive performance (by accuracy of linear SVM) of the selected subsets by the conventional and Instance Weighting (IW) versions of the SVM-RFE algorithm at each iteration of the Recursive Feature Elimination (RFE) process for synthetic data.

on improving the stability and predictive performance of the selected subsets. Figure 3 (left) compares the subset stability (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 3 (right) compares the average classification accuracy (over the 500 training/test trials) of linear SVM at each RFE iteration.

From Figure 3 (left), 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 2), 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 3 (right), we can observe that the subsets selected by **IW** SVM-RFE also result in higher classification 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 Figure 2 and Figure 3 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 classification accuracy based on the selected features.

C. Experiments on Real-World Data

1) Experimental Setup: We experimented with four frequently studied microarray data sets characterized in Table I. 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 I
SUMMARY OF MICROARRAY DATA SETS.

Data Set	# Features	# Instances	Source
Colon	2000	62	[23]
Leukemia	7129	72	[24]
Prostate	6034	102	[25]
Lung	12533	181	[26]

In addition to SVM-RFE and its instance weighting version, **IW** SVM-RFE, we also evaluated the performance of a recently proposed bagging-based ensemble feature selection method [9], using SVM-RFE as the base algorithm. Given a training set, the bagging ensemble method first generates

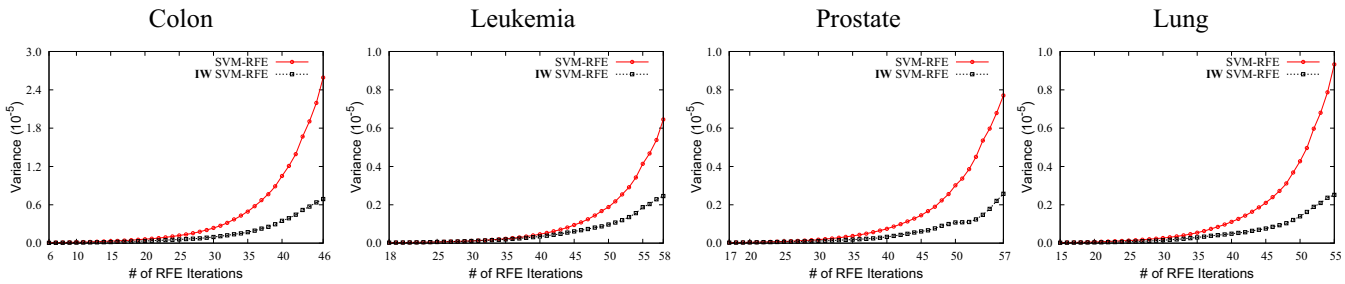

Figure 4. Variance of the feature weights assigned by the conventional and Instance Weighting (**IW**) versions of the SVM-RFE algorithm at each iteration of the Recursive Feature Elimination (RFE) process for microarray data.

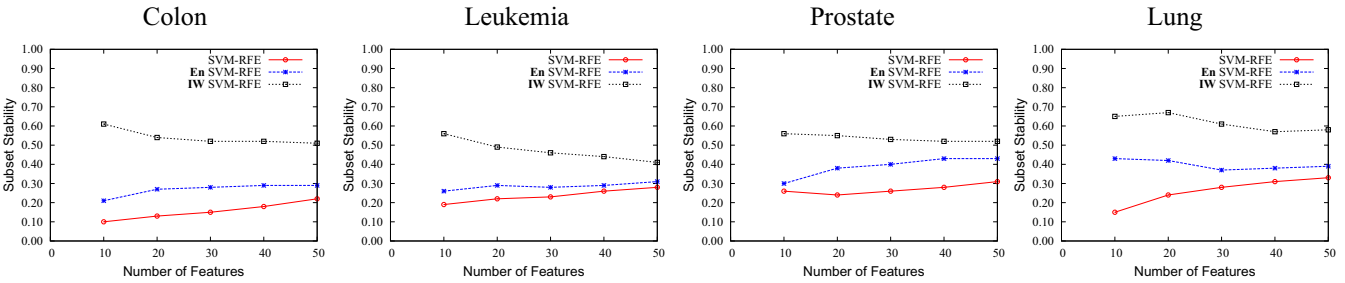

Figure 5. Stability (by Kuncheva Index) of the selected subsets by the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of the SVM-RFE algorithm for microarray data.

a number of bootstrapped training sets, and then repeatedly applies the base algorithm on each of the newly created training sets to generate a number of feature rankings. These rankings are aggregated into a final consensus ranking by summing the ranks of each feature decided based on all bootstrapped training sets. In our implementation, we used 20 bootstrapped training sets to construct the ensemble. We refer to the ensemble version of SVM-RFE as **En** SVM-RFE. In contrast to **IW** SVM-RFE which only applies SVM-RFE once on a weighted training set, **En** SVM-RFE applies SVM-RFE on a number of bootstrapped training sets generated from the original training set.

To evaluate the performance of the three versions of SVM-RFE on a given data set, we applied the 10 fold cross-validation procedure. The original, ensemble, and instance weighting versions of SVM-RFE were 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 was hold out each time. For each selected subset, both a linear SVM and a KNN (K=1) classifiers were trained based on the selected features and the training set, and then tested on the corresponding hold-out test set. The subset stability of each algorithm was measured based on Eq. (8). The predictive performance of each algorithm was measured based on the CV accuracies of the linear SVM and KNN classifiers.

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 study the effect of instance weighting on them 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 4 reports 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 2 for synthetic data). As shown from Figure 4, 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.

3) Stability and Predictive Performance w.r.t. Selected Subsets: Figure 5 reports the subset stability across different numbers of selected features for SVM-RFE in three versions on four data sets. Instance weighting significantly improves the stability of SVM-RFE, which is consistent with both the trend of subset stability improvement observed from synthetic data and the variance reduction effect of instance weighting mentioned 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.

Table II
CLASSIFICATION ACCURACY OF THE SELECTED SUBSETS BY THE CONVENTIONAL, ENSEMBLE (**En**), AND INSTANCE WEIGHTING (**IW**) VERSIONS OF THE SVM-RFE ALGORITHM FOR MICROARRAY DATA.

Data Set	Classifier	Selection Method	Number of Selected Features				
			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

Table II reports the classification accuracy (average value ± standard deviation) of linear SVM and 1NN based on the selected features by the three versions SVM-RFE, respectively. The three algorithms in general lead to very similar classification accuracy. Except for a few cases, the differences in the average accuracy values produced by the three algorithms are insignificant given the standard deviations. The accuracy results in Table II verify that the increased stability resulted from instance weighting (as shown in Figure 5) is not at the price of accuracy.

Observations from Figure 5 and Table II 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 classification accuracy mainly lies in the small sample size of the test sets in microarray data as opposed to synthetic data used in Section V-B. Studying the stability of feature selection provides a new perspective to domain experts in choosing a feature selection algorithm and validating the selected features.

4) Algorithm Efficiency: Figure 6 compares the running time of the three versions of SVM-RFE on the entire data set for each microarray data set. **En** SVM-RFE is almost 20 times slower than SVM-RFE, while **IW** SVM-RFE is only slightly slower than SVM-RFE. The efficiency of **IW** SVM-RFE 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 of **IW** SVM-RFE.

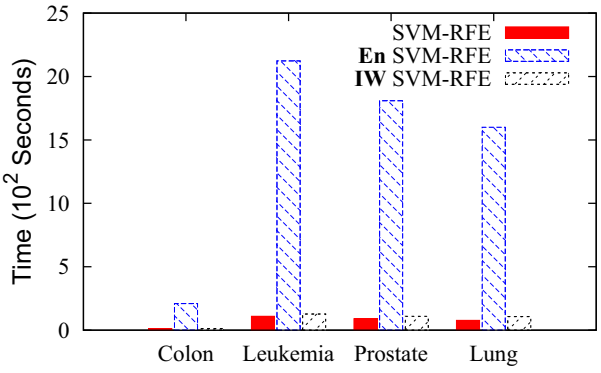

Figure 6. Running time for the conventional, Ensemble (**En**), and Instance Weighting (**IW**) versions of the SVM-RFE algorithm on microarray data.

VI. CONCLUSION

In this paper, we have presented a theoretical framework which reveals the relationship between the stability and accuracy of feature selection. We have also developed an empirical instance weighting framework for variance reduction and a margin based instance weighting algorithm. Our empirical study has verified that instance weighting is an effective and efficient approach to reduce the variance and improve the stability of feature selection algorithms without sacrificing predictive accuracy.

The specific algorithm developed under the instance weighting framework was meant to demonstrate the effectiveness of the framework, and can be improved in various ways. In the future, we plan to investigate alternative methods for weighting instances according to margin vectors and study the effectiveness of the instance weighting framework for other feature selection algorithms. Along the theoretical framework, an interesting direction would be to investigate how the stability of feature selection affects the bias-variance properties of various learning algorithms.

REFERENCES

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

[2] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, vol. 17, no. 4, pp. 491–502, 2005.

[3] M. Wasikowski and X. Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1388–1400, 2010.

[4] M. S. Pepe, R. Etzioni, Z. Feng, et al., "Phases of biomarker development for early detection of cancer," *J Natl Cancer Inst*, vol. 93, pp. 1054–1060, 2001.

[5] C. A. Davis, F. Gerick, V. Hintermair, et al., "Reliable gene signatures for microarray classification: assessment of stability and performance," *Bioinformatics*, vol. 22, pp. 2356–2363, 2006.

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

[7] S. Loscalzo, L. Yu, and C. Ding, "Consensus group based stable feature selection," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-09)*, 2009, pp. 567–576.

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

[9] Y. Saeys, T. Abeel, and Y. V. Peer, "Robust feature selection using ensemble feature selection techniques," in *Proceedings of the ECML Confernce*, 2008, pp. 313–325.

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

[11] B. Cao, D. Shen, J. Sun, Q. Yang, and Z. Chen, "Feature selection in a kernel space," in *Proceedings of the 24th International Conference on Machine learning*, 2007, pp. 121–127.

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

[13] M. Robnik-Sikonja and I. Kononenko, "Theoretical and empirical analysis of Relief and ReliefF," *Machine Learning*, vol. 53, pp. 23–69, 2003.

[14] O. Bousquet and A. Elisseeff, "Stability and generalization," *Journal of Machine Learning Research*, vol. 2, pp. 499–526, 2002.

[15] P. Domingos, "A unified bias-variance decomposition and its applications," in *Proceedings of the Seventeenth Internationl Conference on Machine Learning*, 2000, pp. 231–238.

[16] M. A. Munson and R. Caruana, "On feature selection, bias-variance, and bagging," in *Proceedings of the 20th European Conference on Machine Learning*, 2009, pp. 144–159.

[17] D. Koller and M. Sahami, "Toward optimal feature selection," in *Proceedings of the Thirteenth International Conference on Machine Learning*, 1996, pp. 284–292.

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

[19] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

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

[21] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Computer Systems and Science*, vol. 55, no. 1, pp. 119 – 139, 1997.

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

[23] U. Alon, N. Barkai, D. A. Notterman, et al., "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Natl Acad. Sci. USA*, vol. 96, pp. 6745–6750, 1999.

[24] T. R. Golub, D. K. Slonim, P. Tamayo, et al., "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, 1999.

[25] D. Singh, P. G. Febbo, K. Ross, et al., "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell.*, vol. 2, pp. 203–209, 2002.

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