# Effective Gene Selection Method with Small Sample Sets Using Gradient-Based and Point Injection Techniques

D. Huang and Tommy W.S. Chow

**Abstract**—Microarray gene expression data usually consist of a large amount of genes. Among these genes, only a small fraction are informative for performing a cancer diagnostic test. This paper focuses on effective identification of informative genes. We analyze gene selection models from the perspective of optimization theory. As a result, a new strategy is designed to modify conventional search engines. Also, as overfitting is likely to occur in microarray data because of their small sample set, a point injection technique is developed to address the problem of overfitting. The proposed strategies have been evaluated on three kinds of cancer diagnosis. Our results show that the proposed strategies can improve the performance of gene selection substantially. The experimental results also indicate that the proposed methods are very robust under all of the investigated cases.

**Index Terms**—Gene selection, gradient-based learning, optimization theory, point injection.

## 1 Introduction

Microarray techniques, such as DNA chip and high-density oligonucleotide chip, are powerful biotechnological means because they are able to record the expression levels of thousands of genes simultaneously [7]. Systematic and computational analysis on microarray data enables us to understand phenological and pathologic issues in a genomic level [7], [6]. Microarray data, however, always contains a huge gene set of up to thousands and a small sample set that is down to tens. Moreover, only a very small fraction of the genes are informative for a certain task [8], [18]. For example, Singh et al. [26] used the HG U95A array of Affymetrix as a platform to record the expression profiles of about 100,000 genes for studying prostate cancer. The biological studies summarized by SuperArray Bioscience Corporation http://www.superarray.com/gene_array_product/HTML/OHS-403.html) suggested that only hundreds of genes were (suspiciously) biomarkers of prostate cancer. Different diseases are related to different gene sets. Intuitively, research on the identification of cancer-causing genes has become very challenging. Effectively tackling this problem has many merits. Using a small gene set, we can conduct computational data analysis in a relatively low-dimensional data domain. This is very useful for delivering precise, reliable, and interpretable results. Also, with the gene selection results, biology researchers can focus on only the marker genes and confidently ignore the irrelevant genes. The cost of biological experiment and decision can thus be greatly reduced.

Various machine learning and statistical feature selection models have been directly applied or adapted to gene selection/reduction [3], [8], [10], [14], [20], [17], [25], [29], [30], [32], [33], [35]. A (gene) feature selection framework basically consists of two parts: a search engine to determine the promising feature subset candidates and a criterion to determine which candidate is the best [19], [22]. There are several search engines, including ranking, optimal search, heuristic search, and stochastic search. Feature selection models are categorized as filter model, wrapper model, and embedded model according to the type of evaluation criterion.

The filter selection model that uses heuristic/stochastic search engines ([20], [25], [30], [32]) and the embedded selection model ([17], [10], [33]) are the two widely used feature selection frameworks for conducting gene selection. In this paper, we focus on the former one. A typical filter model is selected and implemented to demonstrate the capabilities of our proposed strategies. In this model, the employed search engine is the sequential forward search (SFS). The evaluation criterion is based on Bayesian discriminant [13].

In a gene (feature) selection scheme, the evaluation criterion and search engine play equally important roles. Evaluation criteria have been heavily investigated in many studies [3], [8], [10], [17], [18], [20], [25], [29], [30], [32], [33], [35]. In contrast, study on search engines has drawn little attention. The heuristic search engines, especially the sequential forward/backward search (SFS/SBS), and the stochastic search engines (e.g., genetic algorithm) are pervasively employed for gene selection. Recent studies reported on the improvement on the searching efficiency of stochastic algorithms [25], [30]. There are, however, few attempts to modify heuristic search algorithms. In the stepwise strategy [24], that is, the floating (compound) search, selecting $k$ features (genes) is followed by eliminating $j$ "worst" selected ones, where $j$ is less than $k$. Al-Ani

- *The authors are with the Electrical Engineering Department, City University of Hong Kong, Hong Kong. E-mail: dihuang@ee.cityu.edu.hk, eetchow@cityu.edu.hk.*

*Manuscript received 2 Dec. 2005; revised 20 July 2006; accepted 13 Sept. 2006; published online 10 Jan. 2007.*
*For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0137-1205.*
*Digital Object Identifier no. 10.1109/TCBB.2007.1021.*

1545-5963/07/$25.00 © 2007 IEEE Published by the IEEE CS, CI, and EMB Societies & the ACM

and Deriche [1] used only the "elite" selected features to identify the important items from unselected features. It is noted that these algorithms are totally designed in a discrete feature (gene) space and the results require the testing of more feature (gene) combinations. The testing of more feature combinations usually leads to an increase in computational effort.

Overfitting is a major issue affecting the performance of final results. In brief, overfitting means that learning results (i.e., the selected gene subsets in this study) may perform perfectly on training data, but are unable to handle testing data satisfactorily. Due to the nature of having small pattern sets, the problem of overfitting is exacerbated in most microarray-based data. In order to alleviate this problem, the use of models with high capability of generalization, such as support vector machine and penalized Cox regression model, has been suggested [10], [11]. But, purely relying on learning algorithms is not enough to tackle overfitting in most cases. It is clear that there is a need to develop a specific strategy to solve the problem of overfitting. Zhou et al. [35] employed a bootstrap framework to obtain reliable mutual information estimates. However, its large computational requirement, which is arguably the main shortcoming of the approach, substantially restricts its application.

In this study, we propose two strategies to address the aforesaid problems. The first strategy is designed to enhance the effectiveness of searching. The second one addresses the problem of overfitting. In order to enhance the effectiveness of searching mechanisms, we analyze the search engines from the perspective of optimization theory [2]. The analysis of this type, which has been overlooked in most previous studies, can reveal a major drawback of sequential search. It is found that conventional search engines cannot perform optimization in a maximal way because their searching mechanisms do not completely incorporate into optimization theory. To address this drawback, we come to optimization theory to formulate a modified strategy. In this strategy, gene selection can then be conducted along the steepest possible optimization direction. To overcome the problem of overfitting means identifying the gene subsets with high generalization capability. A point injection approach is designed. The concept of the injection approach is to generate a number of points according to the distribution of given samples. Then, gene subsets can be assessed using the generated points and the original samples,. In this study, it is proven that this approach has been very effective in tackling the problem of overfitting experienced in gene selection.

The presentation of this paper is organized as follows: In Section 2, a typical sequential forward search (SFS) gene selection model is briefly discussed. Our modified strategies are detailed in Section 3. In Section 4, simulation examples are presented and discussed.

## 2 Sequential Forward Gene Selection Process

Assume that we have a classification data set

$$D = \{X, C\} = \{(x_1, c_1), (x_2, c_2), \dots, (x_N, c_N)\}.$$

$(x_i, c_i)$ represents a data sample,in which $x_i$ is the input vector, and $c_i$ records the class label of $x_i$. $x_i$ is an M-dimensional vector, that is, a sample is described with the expression levels of M genes. We represent these genes as $F = \{f_1, f_2, \dots, f_M\}$. Moreover, the samples in $D$ are grouped into $L$ classes, denoted as $\{\omega_1, \dots \omega_L\}$. For a data sample (say, $x_i$), we have $c_i = \omega_k$, where $1 \leq k \leq L$. A gene subset evaluation criterion is represented by $\Phi(S)$, where $S$ is a gene subset. Furthermore, without loss of generality, we suppose that a large value of $\Phi(S)$ means a good $S$. Thus, the goal of a gene selection process is to maximize $\Phi(S)$ through adjusting $S$. In general, $\Phi(S)$ is optimized in the following ways: After a pool of gene subsets is suggested according to certain rules, $\Phi(S)$ of each suggested subset is calculated. The one with the optimal $\Phi(S)$ is selected out. The schemes for determining the gene subset pools require trading the quality of optimization results with computational consumption. Among these schemes, the SFS strategy is the most popular one. The steps are summarized as follows:

**Step 1.** Set the selected gene subset $S$ to empty.

**Step 2.** Repeat the following until certain stopping conditions are met: Identify the most useful gene (say, $g_u$) from the unselected genes and place it into S. $g_u$ satisfies $g_u = \arg\max_g \Phi(S + g)$.

In this study, a Bayesian discriminant-based criterion (BD) [13] is employed as $\Phi(S)$. With the data set $D$, BD is defined as

$$BD(S) = \frac{1}{N}\sum_{i=1}^{N} \log\frac{p_S(c_i|x_i)}{p_S(\overline{c_i}|x_i)} = \frac{1}{N}\sum_{i=1}^{N} \log\frac{p_S(c_i|x_i)}{1 - p_S(c_i|x_i)}, \quad (1)$$

where $\overline{c_i}$ means all of the classes but class $c_i$ and $p_S(.)$ represents a probability density function estimated based on the gene set $S$. In order to estimate the posterior probabilities $p(c|x)$ in (1), the margin probability $p(x)$ and the joint probability $p(x, c)$ should be obtained first. We use Parzen window [23] to build $p(x)$ and $p(x, c)$. Given the aforementioned data set $D = \{X, C\}$, Parzen window estimators are modeled as

$$p(x, c) = \sum_{x_i \in class\ c} p(x_i)p(x|x_i) = \sum_{x_i \in class\ c} p(x_i)\kappa(x - x_i, h_i), \quad (2)$$

$$p(x) = \sum_{all\ class\ c} p(x, c) = \sum_{all\ x_i \in \mathrm{X}} p(x_i)\kappa(x - x_i, h_i), \quad (3)$$

where $\kappa$ is the kernel function and $h_i$ is the width of $\kappa$. With a proper selection of $\kappa(\cdot)$ and $h$, a Parzen window estimator can converge to the real probability density. We choose the Gaussian function as $\kappa$, that is,

$$\kappa(x - x_\mathrm{i}, h_i) = G(x - x_i, h_i) = \frac{1}{(2\pi h_i^2)^{M/2}} \exp\left(-\frac{1}{2h_i^2}(x - x_i)(x - x_i)^T\right),$$

where $M$ is the dimension of $x$. The width $h_i$ is set with $h_i = 2d(x_i, x_j)$, where $d(x_i, x_j)$ is the euclidean distance between $x_i$ and $x_j$ and $x_j$ is the third nearest neighbor of $x_i$.

Following the general rule, we have $p(x_i) = 1/N$. Thus, according to the Bayes formula, we can model $p(c|x)$ as

$$p(c|x) = \frac{p(x|c)P(c)}{p(x)} = \frac{p(x,c)}{p(x)} = \frac{\sum_{x_i \in \text{ class } c} p(x_i)\kappa(x - x_i, h_i)}{\sum_{\text{all } x_i} p(x_i)\kappa(x - x_i, h_i)}.$$

## 3 Modified Sequential Forward Gene Selection Process

In order to effectively cope with microarray data, we modify the above BD-based sequential forward search scheme (SFS) from two aspects. The first one is a weighting-sample operation, which is derived from optimization theory. The second one is the point injection method designed to address the problem of overfitting.

### 3.1 Weighting-Sample Strategy

The objective of gene selection is to optimize an evaluation criterion. In this study, we use $BD(S)$ as the evaluation criterion. After assuming $f((x,c),S) = p_S(c|x)/(1 - p_S(c|x))$, we have

$$BD(S) = \sum_{all\ (x_i,c_i)} \log(f((x_i,c_i),S)). \tag{4}$$

According to optimization theory, the steepest direction of adjusting $S$ to maximize (4) is

$$\frac{\partial BD(S)}{\partial S} = \sum_{all\ (x_i,c_i)} \frac{\partial BD(S)}{\partial f((x,c),S)} \frac{\partial f((x,c),S)}{\partial S}\bigg|_{(x_i,c_i)}. \tag{5}$$

The above equation indicates that, during the course of optimizing the $BD(S)$, the updating of $S$ depends on $\partial BD(S)/\partial f((x,c),S)$ and $\partial f((x,c),S)/\partial S$. The former one happens in a continuous domain, while the latter one is related to $S$ and has to be tackled in a discrete gene domain. In this sense, (5) cannot be solved directly. In order to maximize the $BD(S)$, many searching schemes have been designed. For example, the SFS tests all combinations of $S$ and an unselected gene. It then identifies the one having the maximal $BD$. Clearly, the SFS is conducted just in a gene domain. In other words, only the second term of (5) is considered by conventional SFS. As a result, the searching direction of the SFS cannot comply with the steepest optimization direction. This shortcoming degrades the effectiveness of optimization.

In order to conduct gene selection along the optimization direction, (5) is considered. To fix the second term of (5), we use a conventional SFS. The first term of (5) can be directly calculated as

$$\frac{\partial BD(S)}{\partial f((x,c),S)} = \frac{1}{f((x,c),S)} = \frac{1 - p_S(c|x)}{p_S(c|x)}. \tag{6}$$

Given $S$, $\partial BD(S)/\partial f((x,c),S)$ is only related to $x$. With this observation, we consider $\partial BD(S)/\partial f((x,c),S)$ as a penalty weight to sample. Thus, our modified search engine is the conventional SFS conducted on the weighted samples. The weights of samples are determined by (6).

Assume that the weight assigned to the data sample $(x_i, c_i)$ is $w_i$. With this weighted data set, we adjust the criterion BD (1) as well as the probability estimations (2) and (3) accordingly. In detail, we have

$$\begin{aligned} BD(S) &= \frac{1}{N}\sum_{i=1}^{N} w_i \log\frac{p_S(c_i|x_i)}{1 - p_S(c_i|x_i)} \\ &= \frac{1}{N}\sum_{i=1}^{N} w_i \log\frac{p_S(x_i,c_i)}{p_S(x_i) - p_S(x_i,c_i)}, \end{aligned} \tag{7}$$

where

$$p(x,c) = \sum_{x_k \in \text{ class } c} \frac{w_k}{N}\kappa(x - x_k, h_k), \tag{8}$$

$$p(x) = \sum_{all\, x_k} \frac{w_k}{N}\kappa(x - x_k, h_k). \tag{9}$$

It is natural that different samples exhibit different contributions to learning processes. Actually, most machine learning algorithms incorporate this idea. For example, when minimizing the mean square error $E = \sum_{\text{all }(x_i,y_i)}(f(x_i,\Lambda) - y_i)^2$ through adjusting the parameter set $\wedge$ of $f$, a steepest decent algorithm [2] was used to determine the direction of the updating $\wedge$ with

$$\begin{aligned} -\frac{\partial E}{\partial \Lambda} &= \sum_{\text{all }(x_i,y_i)} -\frac{\partial E}{\partial f}\frac{\partial f(x,\Lambda)}{\partial \Lambda}\bigg|_{x=x_i,y=y_i} \\ &= \sum_{\text{all }(x_i,y_i)} -(f(x,\Lambda) - y)\frac{\partial f(x,\Lambda)}{\partial \Lambda}\bigg|_{x=x_i,y=y_i}, \end{aligned} \tag{10}$$

where $(x_i, y_i)$ is a given training sample. It is noted that the contribution of $(x_i, y_i)$ is penalized by $|f(x_i,\wedge) - y_i|$. Another example is AdaBoosting [12], which is a typical boosting learning algorithm. During the course of learning, AdaBoosting repeats weighting the sample $(x_i, y_i)$ with $w_i e^{-y_i f(x_i)}$, where $w_i$ is the current weight to $(x_i, y_i)$. Also, in order to reduce the risk of overfitting, it is intuitively expected that the negative samples, that is, incorrectly recognized ones, exhibit more influence on the subsequent learning than the positive ones do. In such a way, the convergence rate can be speeded up and the problem of overfitting can be alleviated [16]. AdaBoosting clearly can meet this expectation and exhibit good performance. Equation (10), however, indicates that the steepest decent algorithm fell short on tackling overfitting in such a way that the correctly recognized patterns still carry large weights. This fact has motivated modifications on the gradient-based algorithms [16]. Consider our proposed pattern-weighting strategy, defined by (6); it penalizes the negative patterns heavily. It will be immensely helpful in alleviating the problem of overfitting.

### 3.2 Point-Injection Strategy

Given a data set $D = \{X, C\}$ drawn from a distribution $\mathbf{P}$ in the data domain $X \times C$. The real goal of the BD-based gene selection process is to maximize $BD_{\mathbf{P}}(S)$. Since $\mathbf{P}$ is unknown in most cases, researchers substitute $BD_{\mathbf{P}}(S)$ with the empirical estimate $BD_{(X,C)}(S)$ (simplified as $BD(S)$, as (1) does). This substitution may have a bias because $BD(S)$ cannot always reflect $BD_{\mathbf{P}}(S)$ correctly. The bias will also lead to overfitting; therefore, the gene subset which is selected to optimize $BD(S)$ cannot optimize the real objective $BD_{\mathbf{P}}(S)$. From the perspective of avoiding overfitting, it is preferable that $BD_{\mathbf{P}}(S)$ vary smoothly around the whole data domain. Samples near each other should correspond to similar performance. This is the rationale behind the point injection technique. This technique is called *noise injection* in much of the literature. But, the newly generated points are not expected to be real noise in this study. We use the term *point injection* instead of *noise injection* throughout the paper to avoid confusion. There are two ways in which the injected points participate in gene selection. With reference to the noise injection approaches for classification training [15], the injected points can be explored completely like the original samples. Using the original samples as well as the injected points, the probability estimation models required by the BD are built and the quality of gene subsets is then evaluated. Alternatively, similarly to the smooth error evaluation schemes [27], the injected points are employed only in the evaluation stage. In other words, the probability estimators are built only upon the given samples. Subsequently, gene subsets are evaluated according to the BD values of the given samples and the injected points. In our study, the latter mechanism is used.

Around a pattern $x_i$, a point injection technique adds $v$ points which are generated from a distribution $b(x - x_i)$. $v$ and $b(x - x_i)$ determine the performance of a point injection scheme [28]. $v$ should be determined in such a way as to strike a balance between performance stability and computational efficiency. Also, it has been argued that, for reasonable choices of $v$, such as $v = 8$, 10, or 20, the effect of point injection is slightly different [15], [28]. We set $v = 10$. As to $b(x - x_i)$, the "width" of $b(x - x_i)$, which determines the variance of the injected points, is crucial. As point injection is used to test the properties of the region around $x_i$, a large width of $b(x - x_i)$ is not expected. On the other hand, a small width of $b(x - x_i)$ must bring an insignificant contribution. To determine an appropriate width, simulation-based strategies can be used [15]; [28]. Also, Sima et al. [27] developed an analytic approach to determine the width of $b(x - x_i)$. Aiming to reduce the bias intrinsic to the resubstitution error estimation as much as possible, this approach relies on the joint distribution $(X, C)$ to determine the width of $b(x - x_i)$. This idea inspires our study.

Suppose that $d_i$ is the distance of $x_i$ to the nearest different-class samples, that is,

$$d_i = \min\|x_i - x_j\|, \quad x_j \notin class\, c_i. \tag{11}$$

We also assume that $d_i/2$ is the boundary of different classes, which means that a point $x'$ should have the same class label as $x_i$ if $||x_i - x'|| \leq d_i/2$. With this assumption, several points around $x_i$ can be generated from the Gaussian distribution $N(x_i, \sigma_i)$. We set $\sigma_i = d_i/6$ to guarantee that $x'$ having $||x_i - x'|| = d_i/2$ occurs with the close-zero probability.

Apart from $N(x_i, d_i/6)$, other distributions are investigated. For example, following [27], we set $\sigma_i$ of $N(x_i, \sigma_i)$ as $d_i/\alpha_i$, where $d_i$ is defined in (11) and $\alpha_i$ is determined by $\alpha_i = F_{D_i}^{-1}(1/2)$, where $F_{Di}$ is the cumulative distribution function of the distance of $x_i$ to the points generated from $N(x_i, \sigma_i)$. We also test the rectangular uniform distribution $b(x - x_i) = R(x_i - d_i/2, x_I + d_i/2,)$. Compared with $N(x_i, d_i/6)$, these distributions produce either similar or inferior results.

### 3.3 The Modified SFS

The computational complexity of the SFS is $O(M^2)$, where $M$ is the number of genes. A microarray gene expression data set generally contains information on thousands or tens of thousands of genes. Clearly, directly handling a huge gene set may be an unbearable computational burden. Given by the fact that most genes originally given in a microarray data set are irrelevant to certain tasks, a widely used prefiltering-gene strategy is used to eliminate the irrelevant or insignificantly relevant genes before the commencement of gene selection. This is an effective way to relieve the computational burden. In detail, for each given gene $g$, $BD(g)$ is calculated based on (7), where $w = 1$ for all samples. The genes with small values of BD are considered irrelevant and eliminated. In such as way, a huge gene set can be safely reduced.

With the proposed point injection and weighting-sample strategies as well as the simple prefilter operation, our modified gene selection procedure can be stated as follows:

**Step 1 (Prefilter).** Calculate BD of each given gene and rank these genes in a descending order of BD. Use the remaining top one-third genes for the following selection process.

**Step 2 (Initialization).** Set the selected gene subset $S$ to empty. Assign a weight of 1 to each sample, that is, $w_i = 1$, $1 \leq i \leq N$. Set the injected point set $\{X', C'\}$ to empty.

**Step 3 (Gene selection).** Repeat the following until one has selected certain genes.

a. From the unselected genes, identify the gene $g_m$ satisfying

$$g_m = \underset{g}{\arg\max}[BD_X(g + S) + BD_{X'}(g + S)]. \tag{12}$$

$BD_X(g + S)$ and $BD_{X'}(g + S)$ are $BD(g + S)$ (7) of the given data $\{X, C\}$ and the injected points $\{X', C'\}$, respectively. We design a particular trick for $BD_{X'}(g + S)$. This trick is to reduce the uncertainty caused by point injection and will be detailed later.

b. (Weighting-sample) Update the sample weights. Set $w_i$ based on (6). Then, normalize $w_i$ as $w_i = w_i / \sum_{i=1}^{N} w_i$.

c. (Point-injection) Set $\{X', C'\}$ with empty. Around each pattern, say $x_i$, produce 10 points based on the distribution $N(x_i, d_i/6)$. Then, place these points into $\{X', C'\}$. These injected points inherit the class label and the weight of $x_i$.

Since the given samples cannot cover the whole data domain, the probability estimators built upon them are not able to describe every part of the data space sufficiently. There are areas where points have small distribution probabilities, that is, $p(x, c_i)$ and $p(x)$ are all small. According to (7), given a point $x$, we have

$$\left|\frac{\partial BD(S)}{\partial p(x, c_i)}\right| \propto \left|\frac{1}{p_S(x, c_i)} - \frac{p(c_i) - 1}{p_S(x) - p_S(x, c_i)}\right|.$$

It shows that, when $p(x, c_i)$ for all $c_i$ are small, a very little change in $x$ may cause an extremely large change in $BD(S)$. It is better to avoid this uncontrollable condition, although it can be argued that the uncertain points equally affect the performance of different gene selection candidates. Using this idea, we calculate $BD_{X'}(g + S)$ as

$$BD_{X'}(g+S) = \frac{1}{|X'|}\sum_{\text{all } x'_i \in X'} w'_i \log\frac{p_{g+S}(c'_i|x'_i)}{p_{g+S}(\overline{c'_i}|x'_i)} \underbrace{\left(\arg\max_{c}(p_{g+S}(x'_i|c))\right)}_{A},$$

where $|X'|$ means the cardinality of $X'$. $c'_i$ and $w'_i$ are the weight and class of $x'_i$. For uncertain points, all of the conditional probabilities must be small. Thus, in the above equation, part A limits the impact of an uncertain point, which is as expected.

## 4 Experimental Results

SFS denotes the conventional BD-based SFS, while our modified SFS is denoted as MSFS. The modified SFS with the maximal-probability-weighting-injected-point strategy is represented by the WMSFS. Prior to gene selection, each gene variable is preprocessed with zero mean and unit variance. Our focus will only be placed on comparing the SFS with the modified SFSs. But, detailed comparative performance between the BD-based gene (feature) selection models and other related methods, for instance, mutual information-based ones, the SVM-based approach, and the distribution-based method, can be found in [13].

### 4.1 Synthetic Data

Our studies begin on a set of 3-class and 8-gene data. The first four features are generated according to:

$$\text{Class } 1 \sim N((1, 1, -1, -1), \sigma),$$
$$\text{Class } 2 \sim \mathrm{N}((-1, -1, 1, 1), \sigma),$$
$$\text{Class } 3 \sim N((1, -1, 1, -1), \sigma).$$

The other four genes are randomly determined from normal distribution with zero means and unit variances. Thus, among eight genes, the first four genes are equally relevant to this classification task and the others are irrelevant. All the examined approaches are required to identify four genes. In this study, a selection result is considered correct only when it includes all relevant genes.

In this study, we generate a small-sample-set data—in total, nine samples, among which three samples are from each of three classes. We test different values of $\sigma$. Clearly, the smaller the $\sigma$ is, the simpler the classification problem is and the less likelihood there is that overfitting occurs as a result. Thus, theoretically, the advantage of MSFS and WMSFS should become significant with the increase of $\sigma$. For each $\sigma$, 10,000 data sets are generated. SFS, MSFS and WMSFS run on these data sets and the correct results are counted.

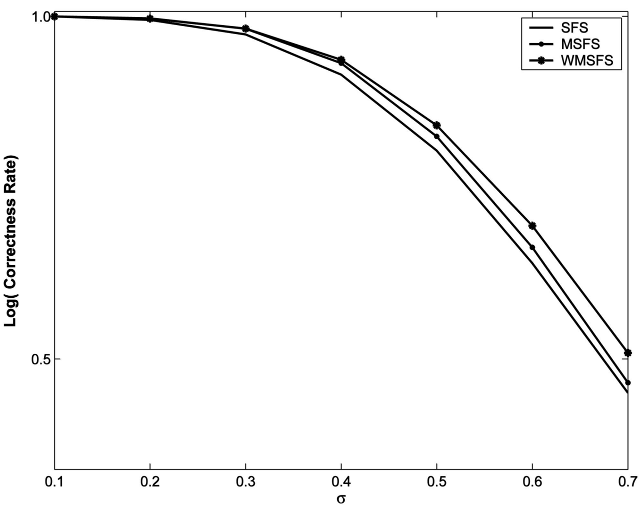

Fig. 1. Comparison between SFS, MSFS, and WMSFS on synthetic data.

The obtained correct rates are illustrated in Fig. 1. These results are consistent with the above theoretical analysis. When $\sigma$ is small, there is an unnoticeable performance difference between the compared methods. Actually, in the case of a small $\sigma$, on the same data set, different methods basically produce the same results. When $\sigma$ is large, however, MSFS and WMSFS illustrate the improved performance. Also, WMSFS is better than MSFS.

### 4.2 Patient Data

Different gene selection methods are compared on several cancer diagnosis data sets. In these real data sets, no a priori knowledge is available. We rely on experimental classification results to assess the quality of gene selection results. Using a selected gene subset, certain classifiers are constructed on training data that are also used for gene selection. Then, we evaluate the built classifiers on the testing data set. Good classification results must indicate a respectable gene subset. We use four typical classifiers, including a multiply percepton model (MLP), two support vector machine models (SVM), and a 3-nearest neighbor rule classifier (3-NN). The MLP used in our study is available at http://www.ncrg.aston.ac.uk/netlab/. For convenience, we set six hidden neurons of MLP for all examples. It is worth noting that a slightly different number of hidden neurons does not have an effect on the overall performance. The number of training cycles is set to 100 to ease the problem of overfitting. The other learning parameters are set with default values. The SVM models used in this study are available at http://www.isis.ecs.soton.ac.uk/resources/svminfo. Two SVM models are employed: the SVM with "Linear" kernel (SVM-L) and the SVM with "RBF" kernel (SVM-R).

#### 4.2.1 *Data Set*

The following gene expression data sets are included in our study.

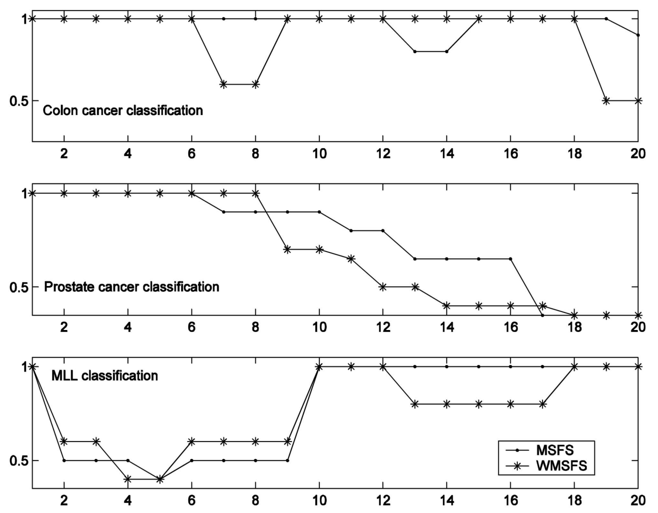

Fig. 2. Statistical results of MSFS and WMSFS on different trials. These results can show the stability of MSFS and WMSFS. The x-axes are the number of the selected genes. The y-axes are LAP.

**Colon tumor classification.** This data contains 62 samples collected from colon cancer patients. Among these samples, 40 samples are tumor and 22 are labeled "normal." There are 2,000 genes selected based on the confidence in the measured expression levels. We split the 62 samples into two disjoint groups—one group with 40 samples for training and the other one with 22 samples for test. We repeat our investigations on 10 different sets of training and testing data to deliver reliable evaluations. The statistical results of 10 trials are presented. Also, in each training data set, the original size ratio between two classes, that is, 40 tumor samples versus 22 normal samples, remains roughly the same.

**Prostate cancer classification.** The objective of this task is to distinguish prostate cancer cases from noncancer cases. The original raw data are published at http://www.genome.wi.mit.edu/mpr/prostate. This data set consists of 102 samples from the same experimental conditions. Each sample is described using 12,600 genes. We split the 102 samples into two disjoint groups—one group with 60 samples for training and the other with 42 samples for testing. Similarly to the last example, the studies on this data are repeated on 10 different sets of training and testing data. The statistical results are summarized and presented in this paper.

**Leukemia subtype classification.** This data, which is available at http://www.broad.mit.edu/cgibin/cancer/datasets.cgi, is used for performing leukemia subtype classification. The given samples are labeled with ALL, MLL, or AML. The training data contains 57 samples—20 labeled with ALL, 17 with MLL, and 20 with AML. In the test data, there are 15 samples—four ALL samples, three MLL ones, and eight AML ones. There are no SVM-related results in this example because the SVM models we employed are designed to deal with 2-class data only.

### 4.2.2 Analysis on the Stability of MSFS and WMSFS

There is an inherent randomness in the course of injecting points. It is thus necessary to investigate the stability of the MSFS and WMSFS. For this purpose, we run the MSFS and WMSFS 10 times on a training data set and compare the obtained results. Given a group of gene subsets of the same size, the appearance probability of each subset is calculated. The largest appearance probability, named LAP, can measure the likelihood of all of the tested subsets being identical. $\mathrm{LAP} = 1$ indicates that all given gene subsets completely match. LAP arrives at its minimum when the tested gene subsets are completely different from each other. In Fig. 2, the results obtained on three data sets are illustrated. It shows that, in most cases, $\mathrm{LAP} = 1$. It means

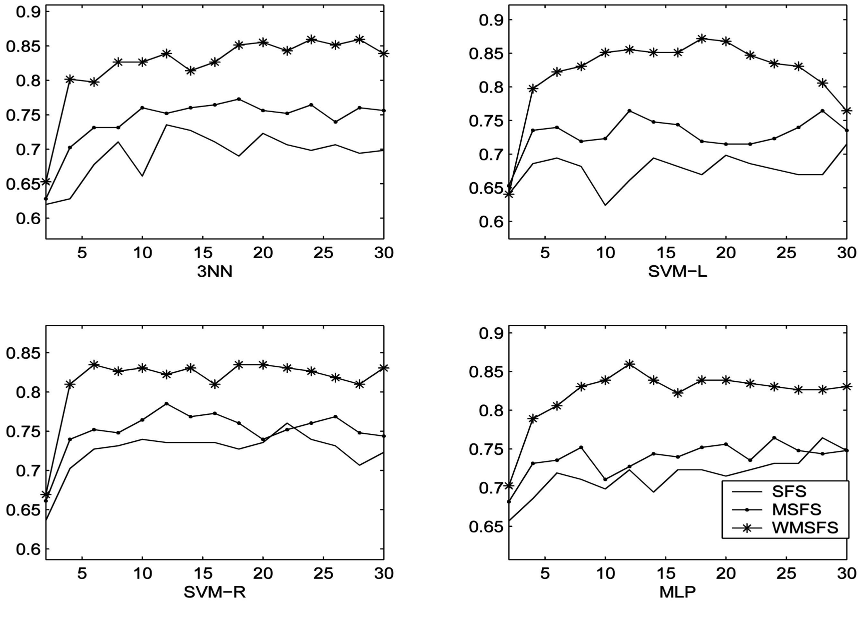

Fig. 3. Comparison on the colon cancer classification data. In these figures, the y-axes are the classification accuracy and the x-axes are the number of the selected genes.

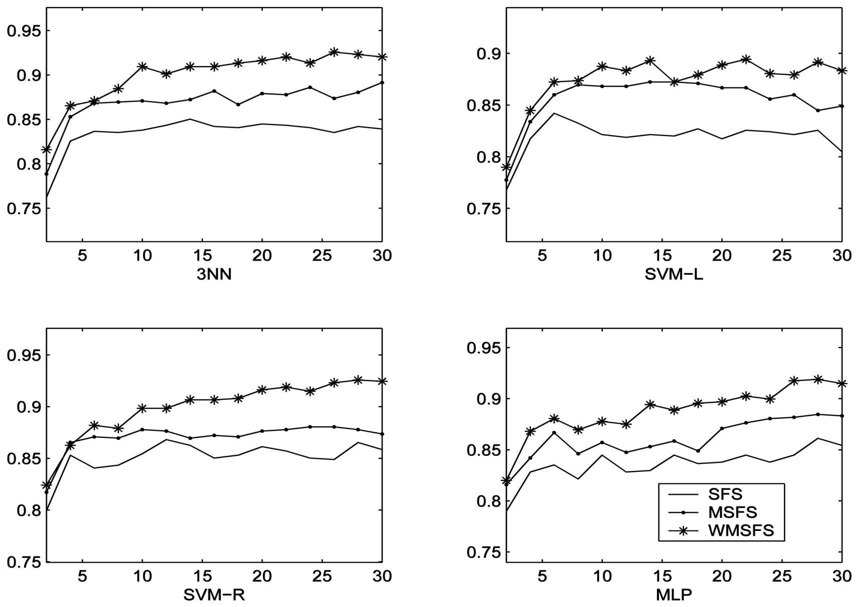

Fig. 4. Comparison on the prostate cancer classification data.

that the MSFS/WMSFS can deliver the same results in different runs using the same training data set.

### 4.2.3 Comparisons of SFS, MSFS, and WMSFS

To demonstrate the merits of our developed strategies, the MSFS and WMSFS are compared with the SFS in terms of classification accuracy. As the MSFS and WMSFS have a stable performance, we run these schemes once on a given data set. The comparative results are presented in Fig. 3 (for colon cancer classification), Fig. 4 (for prostate cancer classification), and Fig. 5 (for leukemia subtype classification).

In most examples, such as the ones about colon cancer and prostate cancer, the proposed MSFS scheme greatly outperforms the conventional SFS. Clearly, this is attributed to the proposed point-injection and sample-weighting strategies. The performance may even be further enhanced when we use the WMSFS in which the influences of uncertain injected points are limited.

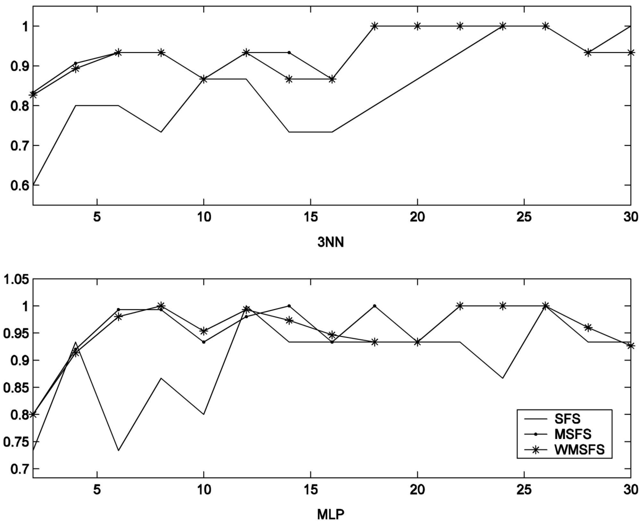

Fig. 5. Comparison on leukemia subtype classification data.

### 4.2.4 Detailed Results on Prostate Cancer

Apart from the above machine learning-based evaluations, we check the obtained results from the biological point of views. In Table 1, one gene result of the WMSFS is listed. The functions of these genes range from cell adhesion (VCL, NELL2) to immune response (DF, C7), from cellular transport (MRC2, RBP1) to regulation of transcription (LMO3), from protein kinase activity (ILK) to hormone activity (IGF1). We note that almost all of these selected genes have been associated with the development and diagnosis of prostate cancer—some of them are well-known prostate cancer-associated genes, such as IGF1, GAGEC1, RBP1, DF, NELL2, ILK, etc., and others have been suggested to overexpress in prostate cancer samples, for example, C7 and LMO3.

## 5 Conclusion

We propose a modified gene selection method and a point injection technique to address the difficulties posed by microarray gene expression data. As the modified gene selection mechanism is designed from the perspective of optimization theory, the overall performance is substantially enhanced. The results obtained on synthetic data and real data demonstrate that the proposed strategies can bring a remarkable improvement on gene selection. Also, we evaluate the two proposed strategies separately. The detailed evaluation results are available at http://www.ee.cityu.edu.hk/~twschow/effective/effective_gene_selection.htm.

The proposed strategies are only applied to one representative gene selection model—BD-based sequential forward search. In future work, we will extend these strategies to other gene selection models and further evaluate their merits and limitations.

TABLE 1
The Genes that Are Identified WMSFS to Be Related to the Prostate Cancer

| Order of selection | Gene symbol | Gene title | Relation with prostate cancer |
|---|---|---|---|
| 1 | VCL | Vinculin | Vinculin, a cytoskeletal protein, can regulate the ability of cancer cell to move away from turmors. It may contribute to metastatic process of prostate cancer. |
| 2 | DF | D component of complement (adipsin) | Adipsin, a member of the trypsin family of peptidases, is a component of the alternative complement pathway playing an important part in humoral suppression of infectious agents.<br>Uzma et al. (2004) find out this gene up-regulates in the samples with prostate diseases, such as prostate cancer. Also, Chow et al. (2001) suggest it a good cancer marker. |
| 3 | MRC2 | Mannose receptor, C type 2 | |
| 4 | NELL2 | NEL-like 2 (chicken) | The close correlation of this gene to prostate cancer is also suggested in other studies (Uzma 2004; Zhang, 2006). |
| 5 | RBP1 | retinol binding protein 1, cellular | Retinoids are involved in cell growth, differentiation, and carcinogenesis. This gene has been found to overexpress in prostate carcinoma (Jerónimo, 2004). |
| 6 | C7 | complement component 7 | This gene takes part in androgen-regulated processes that play important roles in malignant transformation of prostate gland (Uzma, 2004). |
| 7 | IGF1 | homeodomain interacting protein kinase 3 | The role that this gene plays in prostate development and carcinogenesis has been well-recognized and widely examimed (Cheng, 2006). |
| 8 | ILK | integrin-linked kinase | This gene overexpression can suppress anoikis, promote anchorage-independent cell cycle progression, and induce tumorigenesis and invasion (Graff, 2001). |
| 9 | GAGEC1 | G antigen, family C, 1 | The protein encoded in this gene is PAGE4, which is a Cytoplasmic protein and is prostate associated. |
| 10 | LMO3 | LIM domain only 3 (rhombotin-like 2) | The protein encoded in this gene is a LIM-only protein (LMO), which is involved in cell fate determination. This gene has been noted to upregulate in the prostate cancer samples (Uzma 2004). |

## ACKNOWLEDGMENTS

The work described in this paper was fully supported by a grant (SRG-7001846-570) from the City University of Hong Kong.

## REFERENCES

[1] A. Al-Ani and M. Deriche, "Optimal Feature Selection Using Information Maximisation: Case of Biomedical Data," *Proc. 2000 IEEE Signal Processing Soc. Workshop,* vol. 2, pp. 841-850, 2000.

[2] C.M. Bishop, *Neural Networks for Pattern Recognition.* Oxford Univ. Press, 1995.

[3] Y. Chen, E.R. Dougherty, and M. Bittner, "Ratio-Based Decision and Quantitative Analysis of cDNA Microarrays," *J. Biomedical Optics,* vol. 2, pp. 364-374, 1997.

[4] I. Cheng et al., "Common Genetic Variation in IGF1 and Prostate Cancer Risk in the Multiethnic Cohort," *J. Nat'l Cancer Inst.,* vol. 98, no. 2, pp. 123-124, 2006.

[5] M.L. Chow, E.J. Moler, and I.S. Mian, "Identifying Marker Genes in Transcription Profiling Data Using a Mixture of Feature Relevance Experts," *Physiological Genomics,* vol. 5, pp. 99-111, 2001.

[6] S. Dudoit, J. Fridlyand, and T.P. Speed, "Comparison of Discrimination Methods for the Classification of Tumours Using Gene Express Data," *J. Am. Statistical Assoc.,* vol. 97, no. 457, pp. 77-87, 2002.

[7] R. Ekins and F.W. Chu, "Microarrays: Their Origins and Applications," *Trends in Biotechnology,* vol. 17, pp. 217-218, 1999.

[8] T.R. Golub et al., "Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring," *Science,* vol. 286, pp. 531-537, 1999.

[9] J.R. Graff et al., "Integrin-linked Kinase Expression Increases with Prostate Tumor Grade," *Clinical Cancer Research,* vol. 7, pp. 1987-1991, 2002.

[10] I. Guyon, J. Weston, and S. Barnhill, "Gene Selection for Cancer Classification Using Support Vector Machines," *Machine Learning,* vol. 46, pp. 389-422, 2002.

[11] J. Gui and H. Li, "Penalized Cox Regression Analysis in the High-Dimensional and Low-Sample Size Settings, with Application to Microarray Gene Expression Data," *Bioinformatics,* vol. 21, no. 13, pp. 3001-3008, 2005.

[12] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning,* pp. 308-312. Springer, 2001.

[13] D. Huang and T.W.S. Chow, "Efficiently Searching the Important Input Variables Using Bayesian Discriminant," *IEEE Trans. Circuits and Systems,* vol. 52, no. 4, pp. 785-793, 2005.

[14] D. Huang, T.W.S. Chow, E.W.M. Ma, and J. Li, "Efficient Selection of Salient Features from Microarray Gene Expression Data for Cancer Diagnosis," *IEEE Trans. Circuits and Systems, Part I,* vol. 52, no. 9, pp. 1909-1918, 2005

[15] S. Kim et al., "Strong Feature Sets From Small Samples," *J. Computational Biology,* vol. 9, pp. 127-146, 2002.

[16] F. Lampariello and M. Sciandrone, "Efficient Training of RBF Neural Networks for Pattern Recognition," *IEEE Trans. Neural Networks,* vol. 12, no. 5, pp. 1235-1242, 2001.

[17] K.E. Lee et al., "Gene Selection: A Bayesian Variable Selection Approach," *Bioinformatics,* vol. 19, no. 1, pp. 90-97, 2003.

[18] W. Li and Y. Yang, "How Many Genes Are Needed for a Discriminant Microarray Data Analysis?" *Methods of Microarray Data Analysis,* S.M Lin and K.F. Johnson, eds., pp. 137-150, Kluwer Academic, 2002.

[19] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining.* Kluwer Academic, 1998.

[20] X. Liu, A. Krishnan, and A. Mondry, "An Entropy-Based Gene Selection Method for Cancer Classification Using Microarray Data," *BMC Bioinformatics,* vol. 6, no. 76, 2005.

[21] C. Jerónimo et al., "Aberrant Cellular Retinol Binding Protein 1 (CRBP1) Gene Expression and Promoter Methylation in Prostate Cancer," *J. Clinical Pathology,* vol. 57, pp. 872-876, 2004.

[22] L.C. Molina, L. Belanche, and A. Nebot, "Feature Selection Algorithms: A Survey and Experimental Evaluation," technical report, http://www.lsi.upc.es/dept/techreps/html/R02-62.html, 2002.

[23] E. Parzen, "On the Estimation of a Probability Density Function and Mode," *Annals Math. Statistics,* vol. 33, pp. 1064-1076, 1962.

[24] P. Pudil, J. Novovicova, and J. Kittler, "Floating Search Methods in Feature Selection," *Pattern Recognition Letters,* vol. 15, pp. 1119-1125, 1994.

[25] S.C. Shah and A. Kusiak, "Data Mining and Genetic Algorithm Based Gene/SNP Selection," *Intelligence in Medicine,* vol. 31, no. 3, pp. 183-196, 2004.

[26] D. Singh et al., "Gene Expression Correlates of Clinical Prostate Cancer Behavior," *Cancer Cell,* vol. 1, no. 2, pp. 203-209, 2002.

[27] C. Sima, U. Braga-Neto, and E.R. Dougherty, "Superior Feature-Set Ranking for Small Samples Using Bolstered Error Estimation," *Bioinformatics,* vol. 21, no. 7, pp. 1046-1054, 2005.

[28] M. Skurichina, S. Raudys, and R.P. Duin, "K-Nearest Neighbours Directed Noise Injection in Multilayer Perceptron Training," *IEEE Trans. Neural Networks,* vol. 11, no. 2, pp. 504-511, 2000.

[29] Y. Su, T.M. Pavlovic, and S. Kasif, "RankGene: Identification of Diagnostic Genes Based on Expression Data," *Bioinformatics,* vol. 19, no. 12, pp. 1578-1579, 2003.

[30] T.J. Umpai and S. Aitken, "Feature Selection and Classification for Microarray Data Analysis: Evolutionary Methods for Identifying Predictive Genes," *BMC Bioinformatics,* vol. 6, no. 148, 2005.

[31] S.S. Uzma and H.G. Robert, "Fingerprinting the Diseased Prostate: Associations between BPH and Prostate Cancer," *J. Cellular Biochemistry,* vol. 91, pp. 161-169, 2004.

[32] E.P. Xing, M.I. Jordan, and M. Karp, "Feature Selection for High-Dimensional Genomic Microarray Data," *Proc. 18th Int'l Conf. Machine Learning,* 2001.

[33] K. Yeung, R.E. Bumgarner, and A.E. Raftery, "Bayesian Model Averaging: Development of an Improved Multi-Class, Gene Selection and Classification Tool for Microarray Data," *Bioinformatics,* vol. 21, no. 10, pp. 2394-2402, 2005.

[34] C. Zhang et al., "Profiling Alternatively Spliced mRNA Isoforms for Prostate Cancer Classification," *BMC Bioinformatics,* vol. 7, pp. 202-236, 2006.

[35] X. Zhou, X. Wang, and E. Dougherty, "Nonlinear Probit Gene Classification Using Mutual Information and Wavelet-Based Feature Selection," *J. Biological Systems,* vol. 12, no. 3, pp. 371-386, 2004.

**D. Huang** received the PhD degree in 2004 from the Department of Electronic Engineering at the City University of Hong Kong. Her research focuses on machine learning and bioinformatics.

**Tommy W.S. Chow** received the BSc (First Hons) degree. He undertook his Training with Reyrolle Technology, United Kingdom. He received the PhD degree from the University of Sunderland, United Kingdom, working on a collaborative project between International Research and Development, Newcastle upon Tyne, United Kindgom and the Ministry of Defense (Navy) United Kingdom. After receiving the PhD degree, he joined the City University of Hong Kong, where he is currently a professor in the Department of Electronic Engineering. He has been a consultant to the Mass Transit Railway, Kowloon-Canton Railway Corporation, Hong Kong. He has conducted many collaborative projects on the application of neural networks. One of these works also led to the Best Paper Award at the 2002 IEEE Industrial Electronics Society Annual meeting in Seville, Spain. Currently, he is collaborating with the Laboratories of Information Technology, National Singapore University on bioinformatics. His recent research has been in the area of neural network, learning theory, and bioinformatics. He is an author and coauthor of numerous published works, including book chapters, and more than 110 journal articles related to his research areas. He was the chairman of the Hong Kong Institute of Engineers, Control Automation and Instrumentation Division from 1997-1998; he is also a fellow of the IET United Kingdom.