JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 9, Number 1, 2002
Mary Ann Liebert, Inc.
Pp. 127–146

Strong Feature Sets from Small Samples

SEUNGCHAN KIM,[1] EDWARD R. DOUGHERTY,[1] JUNIOR BARRERA,[2]
YIDONG CHEN,[3] MICHAEL L. BITTNER,[3] and JEFFREY M. TRENT[3]

ABSTRACT

For small samples, classifier design algorithms typically suffer from overfitting. Given a set of features, a classifier must be designed and its error estimated. For small samples, an error estimator may be unbiased but, owing to a large variance, often give very optimistic estimates. This paper proposes mitigating the small-sample problem by designing classifiers from a probability distribution resulting from spreading the mass of the sample points to make classification more difficult, while maintaining sample geometry. The algorithm is parameterized by the variance of the spreading distribution. By increasing the spread, the algorithm finds gene sets whose classification accuracy remains strong relative to greater spreading of the sample. The error gives a measure of the strength of the feature set as a function of the spread. The algorithm yields feature sets that can distinguish the two classes, not only for the sample data, but for distributions spread beyond the sample data. For linear classifiers, the topic of the present paper, the classifiers are derived analytically from the model, thereby providing an enormous savings in computation time. The algorithm is applied to cancer classification via cDNA microarrays. In particular, the genes BRCA1 and BRCA2 are associated with a hereditary disposition to breast cancer, and the algorithm is used to find gene sets whose expressions can be used to classify BRCA1 and BRCA2 tumors.

Key words: perceptron, gene expression, classification, cancer.

1. INTRODUCTION

GIVEN A SET OF FEATURES ON WHICH TO BASE A CLASSIFIER, two issues must be addressed: 1) design of a classifier from sample data that is close to optimal; 2) estimation of the error of the designed classifier. Here we are interested in feature selection from a large set of potential features. The key issue is whether a particular feature set provides good classification. Hence, a main concern is the precision with which the error of the designed classifier estimates the error of the optimal classifier. If the amount of data for both design and error estimation is unlimited, then various methods exist to estimate the optimal error to within any desired precision; however, the problem becomes much more difficult in situations where the

[1]Department of Electrical Engineering, Texas A&M University, College Station, TX 77840.
[2]Departamento de Ciencia de Computacao, Universidade de Sao Paulo, Sao Paulo, Brazil.
[3]Cancer Genetic Branch, National Human Genome Research Institute, National Institutes of Health, Bethesda, MD 20892-4470.

amount of data is very limited. In this case, an error estimator may be unbiased but have a large variance, and therefore often be low. This can produce a large number of variable sets and classifiers with low error estimates. A small sample may yield thousands of variable sets for which the error estimate from the data at hand is zero.

In this paper, we propose a procedure that alleviates this problem by designing classifiers from a probability distribution resulting from spreading the mass of the sample points via a circular distribution to make classification more difficult, while maintaining sample geometry. The algorithm is parameterized by the variance of the circular distribution. By considering increasing variances, the algorithm finds gene sets whose classification accuracy remains strong relative to greater spreading of the sample. The error then gives a measure of the strength of the feature set as a function of the variance.

The immediate application of interest is classification via cDNA microarrays, which provide expression measurements for thousands of genes simultaneously (Schena *et al.*, 1995; DeRisi *et al.*, 1997; Duggan *et al.*, 1999). A key goal for the use of expression data is to perform classification via different expression patterns. A successful classifier provides a list of genes whose product abundance is indicative of important differences in cell state, such as healthy or diseased, or one particular type of cancer or another. Among such informative genes are those whose products play a role in the initiation, progression, or maintenance of the disease. Two central goals of molecular analysis of disease are to use such information to directly diagnose the presence or type of disease and to produce therapies based on the disruption or correction of the aberrant function of gene products whose activities are central to the pathology of a disease. Correction would be accomplished either by the use of drugs already known to act on these gene products or by developing new drugs targeting these gene products. Achieving these goals requires designing a classifier that takes a vector of gene expression levels as input and outputs a class label, which predicts the class containing the input vector. Classification can be between different kinds of cancer, different stages of tumor development, or many other such differences. Classifiers are designed from a sample of expression vectors. This requires assessing expression levels from RNA obtained from the different tissues with microarrays, determining genes whose expression levels can be used as classifier variables, and then applying some rule to design the classifier from the sample microarray data. Design, performance evaluation, and application of classifiers must take into account randomness arising from both biological and experimental variability. To rapidly move from expression data to diagnostics that can be integrated into current pathology practice or to useful therapeutics, expression patterns must carry sufficient information to separate sample types. Further, sufficient information must be vested in sets of genes small enough to serve as either convenient diagnostic panels or as candidates for the very expensive and time-consuming analysis required to determine if they could serve as useful targets for therapy.

The inherent power of expression data to separate sample types was first clearly demonstrated by clustering samples on the basis of gene expression patterns. Such demonstrations provided separation, but utilized large numbers of genes: rhabdomyosarcoma, 495 genes (Khan *et al.*, 1998); colon cancer, 2,000 genes (Alon *et al.*, 1999); lymphoma, 4,026 genes (Alizadeh *et al.*, 1999); breast cancer, 1,753 genes (Perou *et al.*, 2000); and melanoma, 3,613 genes (Bittner *et al.*, 2000). Classification using a variety of methods has been used to exploit the class-separating power of expression data using fewer genes: leukemias, 50 genes (Golub *et al.*, 1999); various cancers, 173–4,375 genes (Ben-Dor *et al.*, 2000); small, round, blue-cell cancers, 96 genes (Khan *et al.*, 2001). Even these gene sets are too large to allow construction of a practical immunohistochemical diagnostic panel.

The problem at this stage is that there is a very large set of gene-expression profiles (features) and typically a small number of microarrays (sample points), making it difficult to find the best features from which to construct a classifier. Thus, it behooves us to find gene sets that can perform accurate classification in distributional settings whose dispersions are in excess of the sample data. We will demonstrate the methodology by finding genes and sets of genes that show strong potential for discrimination between types of hereditary breast cancer.

2. CLASSIFIER DESIGN

Given a set of features (random variables) $X_1, X_2, \ldots, X_d$ from which to form a feature vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)$, a binary classification problem involving $\mathbf{X}$ is determined by a binary random variable

Y, taking the values (*class labels*) 0 and 1. A *classifier* ψ is a function of $\mathbf{X}$ for which $\psi(\mathbf{X})$ is an estimator of Y. The error of ψ is defined by the expected absolute difference between Y and $\psi(\mathbf{X})$, $\varepsilon[\psi] = E[|Y - \psi(\mathbf{X})|]$; $\varepsilon[\psi]$ also equals the probability, $P(\psi(\mathbf{X}) \neq Y)$, of an incorrect classification. An optimal classifier, ψ_d, is a function on Euclidean space $\Re^d$ that has minimal error, ε_d, called the *Bayes error*. An optimal classifier is determined by the conditional probability of the label given the observation: $\psi_d(\mathbf{x}) = 1$ if and only if $P(Y = 1|\mathbf{x}) > 0.5$.

Unless the distribution of $(\mathbf{X}, Y)$ known, which is rare, a classifier ψ_n is designed from sample data pairs by a classification rule to estimate ψ_d, and ε_d is estimated by the error ε_n of ψ_n. Since ε_d is the error of the optimal classifier, $\varepsilon_n \geq \varepsilon_d$. There is a *design cost* $\Delta_n = \varepsilon_n - \varepsilon_d$. Since they depend on the sample, ε_n and Δ_n are random variables dependent on the classification rule and the feature-label distribution. A classification rule is *consistent* for the distribution of $(\mathbf{X}, Y)$ if $E[\Delta_n] \to 0$ as $n \to \infty$, where the expectation is relative to the distribution of the sample. If $E[\Delta_n] \to 0$ for any distribution, then the classification rule is *universally consistent*. While theoretically useful and pertinent for large samples, consistency is of little importance for very small samples.

To reduce design cost, one can restrict the functions from which an optimal classifier can be chosen to a function class $\mathcal{C}$. This leads to estimating the optimal *constrained* classifier, $\psi_{\mathcal{C}} \in \mathcal{C}$, having error $\varepsilon_{\mathcal{C}}$. Constraining the classifier reduces design cost at the cost of increasing the error of the best possible classifier. Since optimization in $\mathcal{C}$ is over a subclass of classifiers, $\varepsilon_{\mathcal{C}} \geq \varepsilon_d$. The *constraint cost* is $\Delta_{\mathcal{C}} = \varepsilon_{\mathcal{C}} - \varepsilon_d$. A classification rule yields a classifier $\psi_{n,\mathcal{C}} \in \mathcal{C}$ with error $\varepsilon_{n,\mathcal{C}}$, and $\varepsilon_{n,\mathcal{C}} \geq \varepsilon_{\mathcal{C}} \geq \varepsilon_d$. Design cost for constrained classification is $\Delta_{n,\mathcal{C}} = \varepsilon_{n,\mathcal{C}} - \varepsilon_{\mathcal{C}}$. For small samples, this can be substantially less than Δ_n, depending on $\mathcal{C}$ and the rule. The error of the designed constrained classifier is decomposed as

$$\varepsilon_{n,\mathcal{C}} = \varepsilon_d + \Delta_{\mathcal{C}} + \Delta_{n,\mathcal{C}}. \tag{1}$$

The expected error of the designed classifier from $\mathcal{C}$ is

$$E\left[\varepsilon_{n,\mathcal{C}}\right] = \varepsilon_d + \Delta_{\mathcal{C}} + E\left[\Delta_{n,\mathcal{C}}\right]. \tag{2}$$

The constraint is beneficial if and only if $\Delta_{\mathcal{C}} < E[\Delta_n] - E\left[\Delta_{n,\mathcal{C}}\right]$. If the cost of constraint is less than the decrease in expected design cost, then the expected error of $\psi_{n,\mathcal{C}}$ is less than that of ψ_n.

The use of strong classifiers, as discussed in this paper, applies to any classification rule; however, we focus on perceptrons owing to the small amount of data they require for design relative to more general classifiers. They also have a number of attractive properties: simplicity, a linear-like structure, and contributions of individual variables that can be easily appreciated. For a feature vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)$, a perceptron is defined by

$$\psi(\mathbf{X}) = \mathrm{T}(a_0 + a_1 X_1 + a_2 X_2 + \cdots + a_d X_d) \tag{3}$$

where T is a threshold function, $\mathrm{T}(z) = 0$ if $z \leq 1/2$, and $\mathrm{T}(z) = 1$ if $z > 1/2$. A perceptron splits $\Re^d$ into two by the hyperplane defined by setting the sum in the preceding equation to 0. Design of a perceptron requires estimating the coefficients $a_1, a_2, \ldots, a_d$, and a_0.

To give some idea of our reason for focusing on perceptrons, we consider the alternative of using neural networks, which are multilayer perceptrons. A basic two-layer neural network takes the outputs of k perceptrons (*neurons*) and inputs these outputs into a final perceptron. By increasing the number of neurons, one can arbitrarily decrease the constraint (Cybenko 1989; Funahashi, 1989; Hornik *et al.*, 1989). But this raises the dilemma of balancing the contributions to $E[\varepsilon_{n,\mathcal{C}}]$ in Equation 2. The data requirement grows rapidly as the number of neurons is increased. The advantage in design error of perceptrons over neural networks can be measured by the *Vapnik-Chervonenkis (VC) dimension*, $V_{\mathcal{C}}$, of a constraint, whose definition we leave to the literature (Vapnik *et al.*, 1971; Devroye *et al.*, 1996). The VC dimension grows with diminishing constraint. Generally speaking, the sample size must significantly exceed the VC dimension to have small design error. This is evidenced by a well-known bound for design error: if the

designed classifier is chosen from $\mathcal{C}$ according to which classifier in $\mathcal{C}$ makes the minimum number of errors on the sample data (the empirical-error rule), then

$$E\left[\Delta_{n,\mathcal{C}}\right] \leq 4\sqrt{\frac{V_c \log n + 4}{2n}}. \tag{4}$$

The VC dimension of the perceptron of Equation 3 is $d+1$. The VC dimension of a neural network with k neurons exceeds kd if k is even and exceeds $(k-1)d$ if k is odd (Baum, 1988).

3. PROBLEM OF SMALL-SAMPLE CLASSIFIER ERROR ESTIMATION

If there is sufficient data, then it can be split into training and test data to design a classifier and to estimate the error of the designed classifier, respectively. Its estimated error is the proportion of errors it makes on the test data. The estimate is unbiased and its variance tends to zero as $n \to \infty$. When the data is limited and all of it is used to design the classifier, there are several ways to estimate the classifier error. We comment on two of these. The *resubstitution estimate*, $\underline{\varepsilon}_n$, is the fraction of errors made by ψ_n on the sample data. Typically, it is low-biased, meaning $E\left[\underline{\varepsilon}_n\right] \leq E\left[\varepsilon_n\right]$. For small samples, the bias can be severe. For *leave-one-out estimation*, n classifiers are designed from sample subsets formed by leaving out one sample pair. Each is applied to the left-out pair, and the estimator $\hat{\varepsilon}_n$ is $1/n$ times the number of errors made by the n classifiers. Since the classifiers are designed on sample sizes of $n-1$, $\hat{\varepsilon}_n$ actually estimates the error ε_{n-1}. It is an unbiased estimator of ε_{n-1}, meaning that $E\left[\hat{\varepsilon}_n\right] = E\left[\varepsilon_{n-1}\right]$; however, its variance can be substantial for small n (Devroye *et al.*, 1996). There can be a nonnegligible probability that ε_n is greater than ε_d, but that $\hat{\varepsilon}_n$ is significantly smaller than ε_d. Unless one is prudent, this can lead to the erroneous conclusion that both the designed and optimal classifiers perform well, and the concomitant conclusion that the feature set is good, when in fact the feature set is poor.

To illustrate the problem of leave-one-out estimation, we consider histogram rules. For these, $\Re^d$ is partitioned into a disjoint union of cells, and $\psi_n(\mathbf{x})$ is defined to be 0 or 1 according to which is the majority label in the cell. The cells may change with n. They may depend on the sample points, but not on Y. The cubic histogram rule partitions $\Re^d$ into same-size cubes. If the cube edge length approaches 0 and n times the common volume approaches infinity as $n \to \infty$, then the rule is universally consistent. For any partition, there exists a distribution for which (Devroye *et al.*, 1996)

$$E\left[\left|\hat{\varepsilon}_n - \varepsilon_n\right|^2\right] \geq \frac{1}{e^{1/12}\sqrt{2\pi n}}. \tag{5}$$

Since $E[\hat{\varepsilon}_n] = E[\varepsilon_{n-1}]$, the inequality gives an approximate lower bound on the maximum variance of $\hat{\varepsilon}_n - \varepsilon_n$ over all distributions. Lacking distribution knowledge, we confront the possibility of this lower bound. Taking the square root gives a lower bound on the maximum standard deviation. For $n = 25$, this lower bound is 0.1355; for $n = 50$, it is 0.1139. These are not good for a lower bound, even for a worst-case bound. We cannot be sure that we are not doing worse, perhaps substantially worse, than these. An upper bound for all distributions is given by Devroye *et al.* (1996)

$$E\left[\left|\hat{\varepsilon}_n - \varepsilon_n\right|^2\right] \leq \frac{1+6e^{-1}}{n} + \frac{6}{\sqrt{\pi(n-1)}}. \tag{6}$$

This is not encouraging. Taking the square root gives an upper bound on the variance. For $n = 25$, this upper bound is 0.9051; for $n = 50$, it is 0.7401. These are useless. Using leave-one-out estimation to estimate the Bayes error is risky. Even though $\varepsilon_n \geq \varepsilon_d$, there is nonnegligible likelihood that $\hat{\varepsilon}_n$ will be so beneath ε_n that it gives a very optimistic estimate of ε_d.

Given a large set of potential features, it is necessary to find a small subset that provides good classification. Every subset is a potential feature set. For v variables, there are $2^v - 1$ possible feature vectors. The number of possible vectors can be astronomical, and one cannot apply a classification rule to all of these; nonetheless, even if the classes are moderately separated, for small samples there may be thousands of vectors for which $\hat{\varepsilon}_n \approx 0$. It would be wrong to conclude that the Bayes errors of all the corresponding classifiers are small.

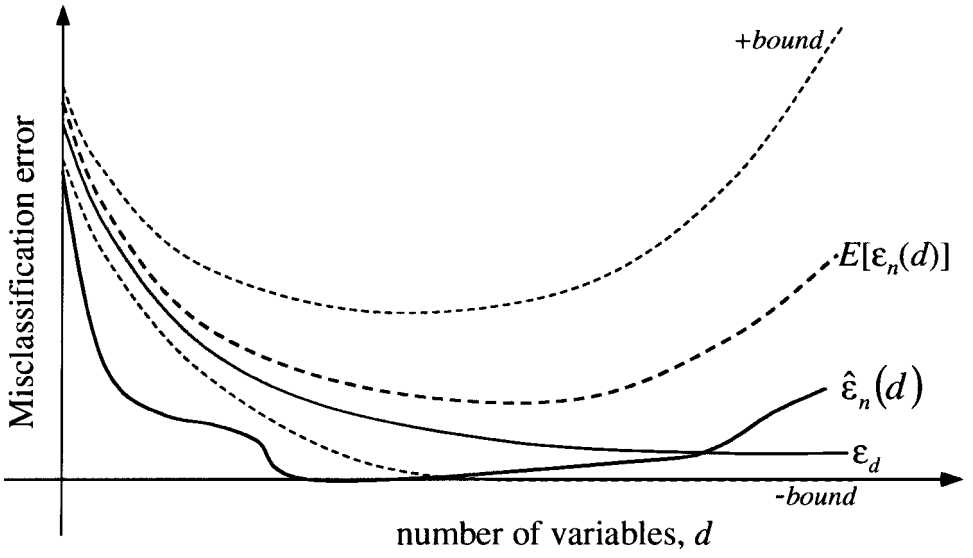


FIG. 1. Estimation of misclassification errors.

Consider the situation in which one has formed a list of variables, and variables are adjoined in a stepwise fashion to the feature vector. As a function of d, the Bayes error decreases, but this is not so for the error of the designed classifier. For fixed sample size n and different numbers of variables d, Fig. 1 shows a generic situation for ε_d and the expected error $E[\varepsilon_n(d)]$ of the designed filter; $\varepsilon_n(d)$ decreases; $E[\varepsilon_n(d)]$ decreases and then increases. Were $E[\varepsilon_n(d)]$ known, then we could conclude that ε_d is no worse than $E[\varepsilon_n(d)]$; however, we have only an estimate of $\varepsilon_n(d)$, which for small samples can be well below (or above) ε_d. Thus, the estimate curve $\hat{\varepsilon}_n(d)$ might drop far below the Bayes-error curve ε_d, even being 0 over a fairly long interval. If we now consider all possible variable subsets of size d, we can expect to have many optimistic estimations.

4. FEATURE STRENGTH

To lower the risk of choosing a feature set based on a low error estimate, rather than design a classifier directly from a small sample, we propose designing it from a distribution based on the sample and for which it is more difficult to distinguish the labels. This will be done in a parameterized manner in which the parameter relates to the difficulty of classification. This paper considers only perceptrons. In principle, the distributional method can be used for other types of classifiers. However, in the perceptron case, we can apply a strictly analytic approach to finding the classifier and its error. This is critical for computation in the context of a large set of features.

To approximate the optimal perceptron, we use the method of finding the optimal mean-square-error (MSE) linear filter and then thresholding. Given the joint feature-label distribution, the optimal linear estimator of Y based on $\mathbf{X}$ is determined by a weight vector $\mathbf{a}$. The autocorrelation matrix for $\mathbf{X}$ and the cross-correlation vector for $\mathbf{X}$ and Y are given by

$$\mathbf{R}_{\mathbf{X}} = \begin{pmatrix} E[X_1X_1] & E[X_1X_2] & \cdots & E[X_1X_d] \\ E[X_2X_1] & E[X_2X_2] & \cdots & E[X_2X_d] \\ \vdots & \vdots & \ddots & \vdots \\ E[X_dX_1] & E[X_dX_2] & \cdots & E[X_dX_d] \end{pmatrix} \tag{7}$$

$$E[\mathbf{X}Y] = \begin{pmatrix} E[X_1Y] \\ E[X_2Y] \\ \vdots \\ E[X_dY] \end{pmatrix}, \tag{8}$$

respectively. If $\mathbf{R_X}$ is nonsingular, then the optimal weight vector is given by $\mathbf{a} = \mathbf{R_X^{-1}} E[\mathbf{X}Y]$. If $\mathbf{R_X}$ is singular, then $\mathbf{R_X^{-1}}$ is replaced by the pseudoinverse of $\mathbf{R_X}$ (Dougherty, 1999). The MSE-based approximation of the optimal perceptron with no constant term is given by $\mathrm{T}(\mathbf{a}^t\mathbf{X})$, where T thresholds at 1/2. The sample-based classification rule for the weight vector is determined by estimating $\mathbf{R_X}$ and $E[\mathbf{X}Y]$. For a given sample $s_n = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^t$, the estimations are given by the matrix and vector

$$\mathbf{R}_{\mathbf{X},n} = \frac{1}{n}\begin{pmatrix} \sum_{k=1}^{n} x_{1k}x_{1k} & \sum_{k=1}^{n} x_{1k}x_{2k} & \cdots & \sum_{k=1}^{n} x_{1k}x_{dk} \\ \sum_{k=1}^{n} x_{2k}x_{1k} & \sum_{k=1}^{n} x_{2k}x_{2k} & \cdots & \sum_{k=1}^{n} x_{2k}x_{dk} \\ \vdots & \vdots & \ddots & \vdots \\ \sum_{k=1}^{n} x_{dk}x_{1k} & \sum_{k=1}^{n} x_{dk}x_{2k} & \cdots & \sum_{k=1}^{n} x_{dk}x_{dk} \end{pmatrix} \tag{9}$$

$$E[\mathbf{X}Y]_n = \frac{1}{n}\begin{pmatrix} \sum_{k=1}^{n} x_{1k}y_k \\ \sum_{k=1}^{n} x_{2k}y_k \\ \vdots \\ \sum_{k=1}^{n} x_{dk}y_k \end{pmatrix}. \tag{10}$$

Since we desire a perceptron with a constant term, we apply the preceding considerations to the augmented vector $(1, \mathbf{X}^t)^t$.

To spread the mass of the given sample s_n, we consider the random vector $(\mathbf{U}, V)$ having the equally likely outcomes $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$. The autocorrelation matrix for $\mathbf{U}$ (not an estimate) is given by $\mathbf{R_U} = \mathbf{R}_{\mathbf{X},n}$, and the cross-correlation vector for $\mathbf{U}$ and V is given by $E[\mathbf{U}V] = E[\mathbf{X}Y]_n$. If $\mathbf{Z}$ is a zero-mean random vector that is independent of $(\mathbf{U}, V)$, we can consider the random vector $(\mathbf{U} + \mathbf{Z}, V)$. By independence,

$$\mathbf{R}_{\mathbf{U}+\mathbf{Z}} = \mathbf{R_U} + \mathbf{R_Z}, \tag{11}$$

$$E[(\mathbf{U} + \mathbf{Z})V] = E[\mathbf{U}V]. \tag{12}$$

Hence, the optimal linear estimator of V in terms of $\mathbf{U}$ has the weight vector

$$\mathbf{w} = (\mathbf{R_U} + \mathbf{R_Z})^{-1} E[\mathbf{U}V] = \left(\mathbf{R}_{\mathbf{X},n} + \mathbf{R_Z}\right)^{-1} E[\mathbf{X}Y]_n. \tag{13}$$

The classifying perceptron is obtained by applying T to the linear estimator determined by $\mathbf{w}$. If $\mathbf{Z}$ is parameterized by its variance σ^2, then the resulting perceptron, ψ_σ, is parameterized by that variance. We call $\mathbf{Z}$ the *spread*, its distribution the *spread distribution*, and ψ_σ the σ-perceptron.

The error, ε_σ, for ψ_σ can be computed analytically from the defining hyperplane. Label the half-spaces determined by the hyperplane as A_0 and A_1, where these refer to the values of $\mathbf{x}$ that are 0 and 1, respectively, for the classifier. Then, for $\mathbf{Z}$ with variance σ^2,

$$\varepsilon_\sigma = \frac{1}{n}\left(\sum_{\{\mathbf{x}_k : y_k = 1\}} \int_{A_0} f_{\mathbf{Z}}(\mathbf{z} - \mathbf{x}_k)\mathrm{d}\mathbf{z} + \sum_{\{\mathbf{x}_k : y_k = 0\}} \int_{A_1} f_{\mathbf{Z}}(\mathbf{z} - \mathbf{x}_k)\mathrm{d}\mathbf{z}\right). \tag{14}$$

For fixed σ, if ε_σ is treated as a function of d by adjoining variables to the feature vector, then ε_σ is a decreasing function of d because it is computed analytically from the distribution, not estimated from a sample. For $\sigma = 0$, which means there is no spreading of the sample mass, ε_σ is equal to the resubstitution error estimate for the sample. Figure 2 illustrates the effects of increasing σ on the error under the assumption that the noise vector possesses a uniform circular distribution. Dots and diamonds denote labels 0 and 1, respectively. We have used both uniform circular and uncorrelated Gaussian distributions. The autocorrelation matrices for these are $\mathbf{R_Z} = r^2(d+2)^{-2}\mathbf{I}$ and $\mathbf{R_Z} = \sigma^2\mathbf{I}$, respectively, where r is the circle radius and $\mathbf{I}$ is the identity matrix. Equation 14 is expressed in detail for these distributions in the Appendix.

We define the *strength* of the feature vector $\mathbf{X}$ relative to the sample and spread distribution by

$$\zeta_{\mathbf{X}}(\sigma) = 1 - \varepsilon_\sigma. \tag{15}$$

Strength is a decreasing function of σ. We say that feature vector $\mathbf{X}_1$ is *stronger* than feature vector $\mathbf{X}_2$ at *spread* σ if $\zeta_{\mathbf{X}_1}(\sigma) \geq \zeta_{\mathbf{X}_2}(\sigma)$. Vector $\mathbf{X}_1$ is *uniformly stronger* than $\mathbf{X}_2$ if $\zeta_{\mathbf{X}_1}(\sigma) \geq \zeta_{\mathbf{X}_2}(\sigma)$ for all $\sigma \geq 0$. For a family $\mathcal{F}$ of feature vectors, $\mathbf{X}_0$ is the *strongest* at σ in $\mathcal{F}$ if

$$\zeta_{\mathbf{X}_0}(\sigma) = \sup_{\mathbf{X}\in\mathcal{F}} \zeta_{\mathbf{X}}(\sigma). \tag{16}$$

It is *uniformly strongest* in $\mathcal{F}$ if the equality holds for all $\sigma \geq 0$.

To examine the behavior of the designed classifier and the error estimator relative to σ, we consider a two-variable model in which each class is defined by an uncorrelated Gaussian distribution with variances 1, and the means of the two classes are separated in such a way that the optimal linear classifier relative to the distributions has Bayes error, ε_{opt}, 0.0786. We let the total sample size be $n = 20$ and compute the σ-perceptron for various sizes of σ between 0 and 1.2. This is done randomly 1,000 times to estimate several error curves: 1) the expected σ-error, $E[\varepsilon_\sigma]$; 2) the expected leave-one-out error, $E[\hat{\varepsilon}_n(\sigma)]$, for the σ-perceptron; and 3) the expected leave-one-out error, $E[\hat{\varepsilon}_{\text{HK}}]$, for the Ho-Kashyap classifier on the sample data (which is not dependent on σ). The curves are shown in Fig. 3(a) along with one-standard-deviation error bars for ε_σ, $\hat{\varepsilon}_n(\sigma)$, and $\hat{\varepsilon}_{\text{HK}}$, and the Bayes error. The curve for $E[\varepsilon_n(\sigma)]$ is not shown because it is very close to $E[\hat{\varepsilon}_n(\sigma)]$, since $E[\hat{\varepsilon}_n(\sigma)] = E[\varepsilon_{n-1}(\sigma)]$. Moreover, we have not shown $E[\varepsilon_{\text{HK}}]$, the expected error for the Ho-Kashyap classifier, since it is very close to $E[\hat{\varepsilon}_{\text{HK}}]$ for the same reason.

From Fig. 3(a), we see that $E[\varepsilon_n(\sigma)] < E[\varepsilon_{\text{HK}}]$ for $0 \leq \sigma \leq 1$, so that in this range the expected performance of the σ-perceptron is better than that of the Ho-Kashyap classifier. For $\sigma = 0.4$, $E[\varepsilon_\sigma]$ is approximately equal to the Bayes error, so that $\varepsilon_{0.4}$ provides an approximately unbiased estimator of the Bayes error. For $\sigma = 0.6$, $E[\varepsilon_\sigma] \approx E[\varepsilon_n(\sigma)]$, so that $\varepsilon_{0.6}$ provides an approximately unbiased estimator of the expected error of the designed σ-perceptron. For $\sigma = 0.8$, the Bayes error is approximately one standard deviation of the σ-error below the mean σ-error, so that $\varepsilon_{0.8}$ provides a conservative estimator of the Bayes error. The danger of using the leave-one-out error is evident from the error bars. Even for $\sigma = 1.2$, the one-standard-deviation bar is well below the Bayes error for both the σ-perceptron and the Ho-Kashyap classifier. The tighter variance of the σ-error as compared to the leave-one-error is demonstrated in Fig. 3(b), which shows standard-deviation curves for the various errors. The lower variance of the true error is of little use in practice, since the true error is not obtainable without knowledge of the distribution.

When designing a classifier from training data, it is necessary to choose values of the spread σ at which to design the classifier. A simple rule of thumb is to set a threshold for the σ-error and push σ as high as possible while keeping the error below the threshold. While this approach can be (and has been) used to find feature sets, it is unsystematic and does not use normalized spread values. A systematic approach is to derive a dispersion value for the sample data and use that value to arrive at normalized spread values. Consider a feature vector $\mathbf{X} = (X_1, X_2, \ldots, X_d)$ and let $\sigma_{k,\mathcal{C}}$ be the standard deviation of X_k on the class $\mathcal{C}$. In practice, $\sigma_{k,\mathcal{C}}$ is estimated from the training data. Define

$$\sigma_{\max} = \max\left\{\sigma_{1,\mathcal{C}_0}, \ldots, \sigma_{d,\mathcal{C}_0}, \sigma_{1,\mathcal{C}_1}, \ldots, \sigma_{d,\mathcal{C}_1}\right\}. \tag{17}$$

A normalized spread, σ_{nor}, between 0 and 1 is chosen. This value corresponds to the situation in Fig. 3 in which all variances are 1. To obtain a corresponding spread for the sample data, we take

$$\sigma = \sigma_{\text{nor}}\sigma_{\max}. \tag{18}$$

By using $\sigma_{\max}$ for the normalization, we maintain a conservative attitude towards estimation of misclassification error. One could perhaps take a less conservative approach if so desired.

5. ALGORITHM FOR SELECTING FEATURE SETS

The task is to find strong feature sets for increasing values of σ. The difficulty is that the search is combinatorial and cannot exhaust all possible feature sets. Hence, a suboptimal search is necessary.

Let $\mathcal{X}_d = \{\mathbf{X}_{d,1}, \mathbf{X}_{d,2}, \ldots, \mathbf{X}_{d,m(d)}\}$ denote the class of all feature vectors of size d that can be constructed from n available features. Then $m(d) = C_{n,d}$, the number of combinations of size d that can be formed from a set of size n. For small n and d, we can try all possible combinations. Even if n is quite large, we may be able to exhaust the full set of combinations for $d = 2$. Beyond that, exhaustion is impossible for large n. Various combinatorial search algorithms and other heuristic algorithms (Srinivas *et al.*, 1994; Bresina, 1996; Li *et al.*, 1998; Mohan. *et al.*, 1999) can be employed, none of which provides an exhaustive search. A simple and quick approach is a random-walk search (Masri *et al.*, 1980; Solis *et al.*, 1981). As often reported, it can provide a good solution in some cases, but may not work well if we are interested in finding as many solutions as possible that satisfy the constraints of the problem. Also, it is not computationally efficient (Ustyuzhaninov, 1980; Rubinstein *et al.*, 1982).

We use a heuristic search algorithm, a kind of *guided random walk* (Price, 1983; Ali *et al.*, 1994; Bresina, 1996; Mohan *et al.*, 1999) that utilizes some probabilistic information constructed from a previous search and evaluation of the error with smaller or the same size d. There are quite a few algorithms available, perhaps the most famous being the *stochastic-based search algorithms* (Hogg, 1996; Mohan *et al.*, 1999) and *genetic search* (Goldberg, 1989; Srinivas *et al.*, 1994).

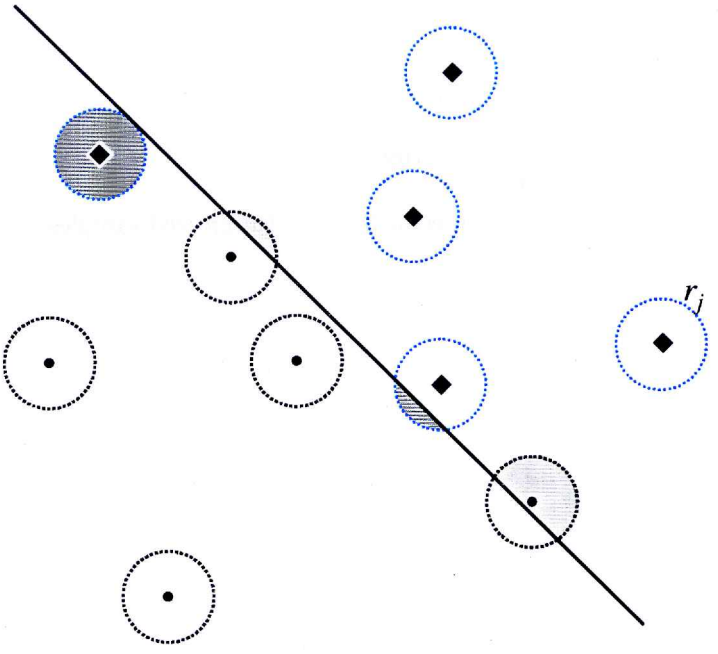
The idea is the following: if a feature is a part of good solution set with small number of features in it, then it is more likely to be a part of a good solution using a larger set, therefore giving more chance to being selected in a feature vector. This can be viewed as a simplest form of genetic algorithm, with only *reproduction.* In another direction, if a good feature set is found in some part of the search space, then we may want to move to another region to find more solutions, the same simple genetic algorithm with a different objective function. This would be useful for finding many *good* solutions rather than finding *a best* solution.

To quantify these considerations, let $\mathcal{X}_{d,X} = \{\mathbf{X}_{d,1}, \mathbf{X}_{d,2}, \ldots, \mathbf{X}_{d,k}\}$ denote the class of all feature vectors that include feature X in them. Then, let $\varepsilon[X]$ be the classification error for X, and

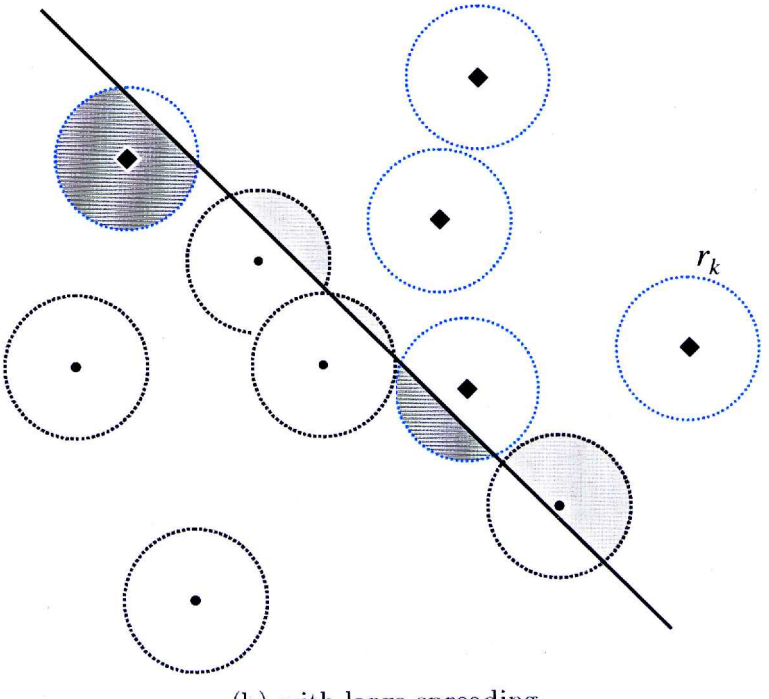
$$\begin{aligned}
\varepsilon_{1,d}[X] &= \max\{\varepsilon[\mathbf{X}] : \mathbf{X} \in \mathcal{X}_{d,X}\} \\
\varepsilon_{2,d}[X] &= \min\{\varepsilon[\mathbf{X}] : \mathbf{X} \in \mathcal{X}_{d,X}\} \\
\varepsilon_{3,d}[X] &= \frac{1}{\nu_d(X)} \sum_{\{\mathbf{X}:\mathbf{X}\in\mathcal{X}_{d,X}\}} \varepsilon[\mathbf{X}]
\end{aligned} \tag{19}$$

where $\nu_d(X)$ is the number of feature sets of size d containing X. In turn, these measures give the maximum, minimum, and average error among feature sets containing X, respectively. We define the *poorness* of a feature X by

$$\eta_d[X] = \sum_{i=1}^{3} \alpha_i \varepsilon_{i,d}[X] \tag{20}$$



(a) with small spreading



(b) with large spreading

FIG. 2. Misclassification error for *distribution*-ized samples.

where α_i is a weight on each error. Then, the poorness of a feature X_k can be normalized as

$$\eta_{d,0}[X_k] = \frac{\eta_d[X_k]}{\sum_{j=1}^{n} \eta_d[X_j]}. \tag{21}$$

The sum of all *normalized poornesses* is 1. Similarly, the *goodness* and *normalized goodness* of X_k are defined as

$$\gamma_d[X_k] = \sum_{i=1}^{3} \alpha_i (1 - \varepsilon_{i,d}[X_k]) \tag{22}$$

$$\gamma_{d,0}[X_k] = \frac{\gamma_d[X_k]}{\sum_{j=1}^{n} \gamma_d[X_j]}. \tag{23}$$

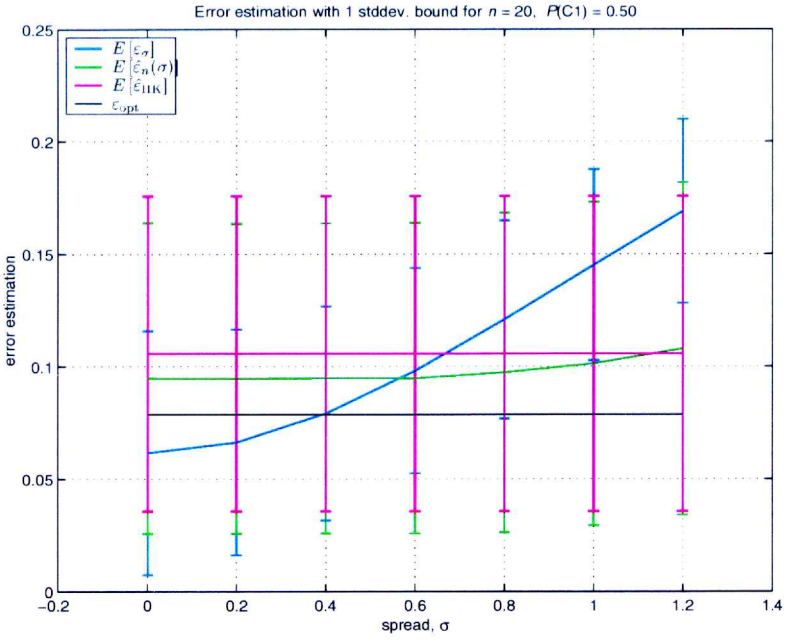
Either the normalized poorness or goodness can be used to construct a feature set. Owing to normalization, either can be used as the probability of a feature being selected in the construction. If all poornesses or goodnesses are equal, then the search is reduced to a regular random walk.

The choice of poorness or goodness depends on the type of feature desired by the algorithm. If conditions are stringent, such as we desire a low error rate and the labels are not well separated, then we may be satisfied in finding a single feature set, in which case we might choose to use goodness as a probability. On the other hand, if conditions are easy and we wish to find many good feature sets, then it may be better to use poorness. The choice of values for each α_i is also an issue. It also depends on the conditions of the problem and the type of solutions desired. A few choices of immediate use are $(0, 1, 0)$, $(0, 0, 1)$, and $(1, 1, 1)$ for $(\alpha_1, \alpha_2, \alpha_3)$. The first alpha vector uses the minimum, the second the expectation, and the third an equal weighting between all. Basically, this is a kind of random walk through space, but with a little bit of help by exploiting the previous walks. Like other combinatorial search algorithms, its performance depends on the appropriateness of the heuristics.

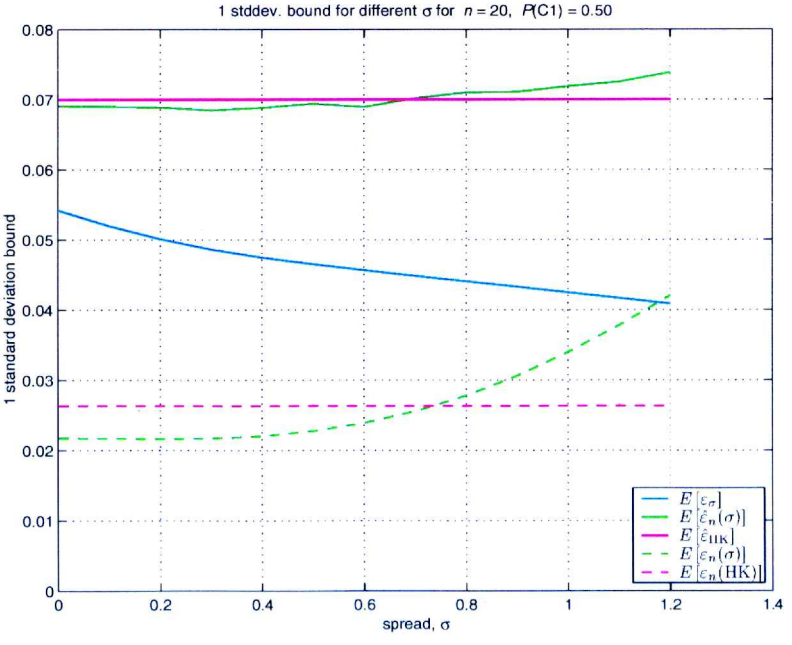
There are many potential variants of the current search algorithm. For instance, we can apply a full genetic search algorithm; however, full implementation of a genetic algorithm can be computationally burdensome. The algorithm has been effective in its present form and useful results have been demonstrated by running it on the Beowulf cluster-based parallel system at CIT/NIH. The algorithm has been designed from the outset so that it can be easily opted to a *parallelization*. It is expected that other search strategies may be applied in the future, depending on the application and the computing environment.

6. APPLICATION TO CLASSIFICATION OF HEREDITARY BREAST CANCER

The scheme for finding strong classifier genes was tested on a published data set (Hedenfalk *et al.*, 2001) comparing the expression profiles of breast tumors from patients carrying mutations in the predisposing genes, BRCA1 or BRCA2, or from patients not expected to carry a hereditary predisposing mutation. Pathological features can help to distinguish BRCA1 and BRCA2 mutation-positive tumors. For BRCA1, there is a higher mitotic count, the presence of pushing tumor margins, and the presence of lymphocitic infiltrate. BRCA2 tumors comprise a more heterogeneous group exhibiting substantially less tubule formation than sporadic breast cancers. BRCA1 associated tumors are generally both estrogen receptor and progesterone receptor negative, while BRCA2 derived tumors are more variable in terms of hormone receptor expression. Altogether, these pathological and genetic differences appear to imply different but overlapping functions for BRCA1 and BRCA2. In the aforementioned study, cDNA microarrays have been used in conjunction with classification algorithms to show the feasibility of using differences in global gene expression profiles to separate BRCA1 and BRCA2 mutation-positive breast cancers. Owing to the small sample sizes involved, gene-expression-profile sets for accurate classification across a large



(a) estimated errors



(b) 1 standard-deviation bounds

FIG. 3. Estimated errors with 1-standard-deviation bounds shown.

population could not be quantitatively discovered (Devroye *et al.*, 1996, Dougherty, 2001). Here we show how the derivation of strong classifiers can lead to the discovery of meaningful feature sets.

Starting with 3,226 genes, we applied the strong-feature algorithm to subsets of size d equal to 1 through 10 to find σ-perceptrons for classification between BRCA1 tumors and the collection of both BRCA2 and sporadic tumors. We used various values of σ between 0.2 and 0.9. Owing to the size of the search space, we exploited the full search space for $d = 1$ and 2 and looked at more than 10,000,000 feature vectors for each $d > 2$. To implement the algorithm, a Beowulf-type cluster-based parallel computer at the Center for Information Technology at the National Institutes of Health was used. The total number of CPUs utilized

was more than 72 and it took about two hours to finish the job. For each d, there is a gene set obtaining minimum error, and in each case a list of outstanding gene sets has been generated. In a joint project, IBM and NuTec Sciences, Inc. are building a 5,000 CPU machine that will be used to apply various microarray-based algorithms. Projections show that the current breast-cancer application can be run in less than one hour using six billion feature sets at each stage, thereby assuring results that are very close to optimal.

The results include many gene pairs that provide good classification for classifying BRCA1 versus the others (BRCA2 and sporadic). Some of these are shown in Table 1, which includes the σ-error for $\sigma_{\text{nor}} = 0.6$ and $\sigma_{\text{nor}} = 0.8$. Strong performing 3-gene classifier sets are also shown in Table 1. In the table, the gene sets are sorted in ascending order for $\sigma_{\text{nor}} = 0.8$. For all classifier sets shown in the tables, the leave-one-out error is 0. This compares favorably to the leave-one-out error obtained in the original study, where 51 genes were used to obtain a leave-one-out error of 1/22. Figure 4 shows the hyperplane constructed with a spread of $\sigma_{\text{nor}} = 0.8$ to classify BRCA1 from BRCA2 and sporadic tissues using the two genes named on the axes. We see that with a significant spread the σ-perceptron yields a very small σ-error and no misclassifications. In the figure, the closest samples from each class are shown with the corresponding distances from the hyperplane.

On the basis of both histology and overall expression profiles, it is easier to separate BRCA1 from BRCA2 and sporadic tumors than to separate BRCA2 and sporadic tumors from each other. This implies a greater difficulty in classifying BRCA2 versus others (BRCA1 and sporadic) than for BRCA1 versus others. This greater difficulty is observed when finding strong classifiers. Table 2 shows strong performing 2- and 3-gene classifier sets. In particular, the 3-gene σ-errors are very small. The leave-one-out error is 0 for all sets listed in the tables. This compares very favorably with the leave-one-out error of 4/22 obtained in the original study using 51 genes.

Another indication of the power of the strong-classifier approach can be seen with regard to 3-class separation (BRCA1, BRCA2, sporadic). In the original study (Hedenfalk *et al*., 2001), 51 genes were used to produce a multidimensional scaling plot to show decent separation of the classes. Perfect class separation is achieved by three genes found using strong-classifier analysis. Figure 5 shows this separation using KRT8, TCF12, and ARVCF. The triple KRT8, TM4SF1, and ARVCF could also be used to show the separation. As noted in the introduction, small gene sets are important for the construction of practical immunohistochemical diagnostic panels.

Table 1. Feature Sets to Classify BRCA1 Tissues Versus BRCA2 and Sporadic Tissues

				Error (%)	
		Feature set		$\sigma_{nor} = 0.6$	$\sigma_{nor} = 0.8$
Two genes	CTPS	EIF2C2		2.7	5.9
	cDNA FLJ13495 fis	BRPF3		4.0	7.1
	cDNA FLJ13495 fis	BRF1		4.4	7.5
	EIF2C2	MSH2		4.3	7.7
	OXCT	cDNA FLJ13495 fis		4.7	8.6
	KRT8	DRPLA		5.2	9.0
	KRT8	ETS2		6.1	9.5
	KRT8	TCF12		7.2	11.8
	MGC1780	MIF		8.6	12.7
	MGC1780	BRPF3		9.1	13.2
	KIAA0020	BRPF3		10.0	14.5
Three genes	CTPS	LOC51723	EIF2C2	2.6	5.7
	CLNS1A	CTPS	EIF2C2	2.7	5.9
	CTPS	GRP58	EIF2C2	3.4	6.9
	MGC1780	SPS	MIF	4.0	7.8
	SIAH1	KRT8	DRPLA	4.9	8.6
	KRT8	PLEC1	DRPLA	4.9	8.8
	CTPS	EIF2C2	FLJ13910	4.6	8.9

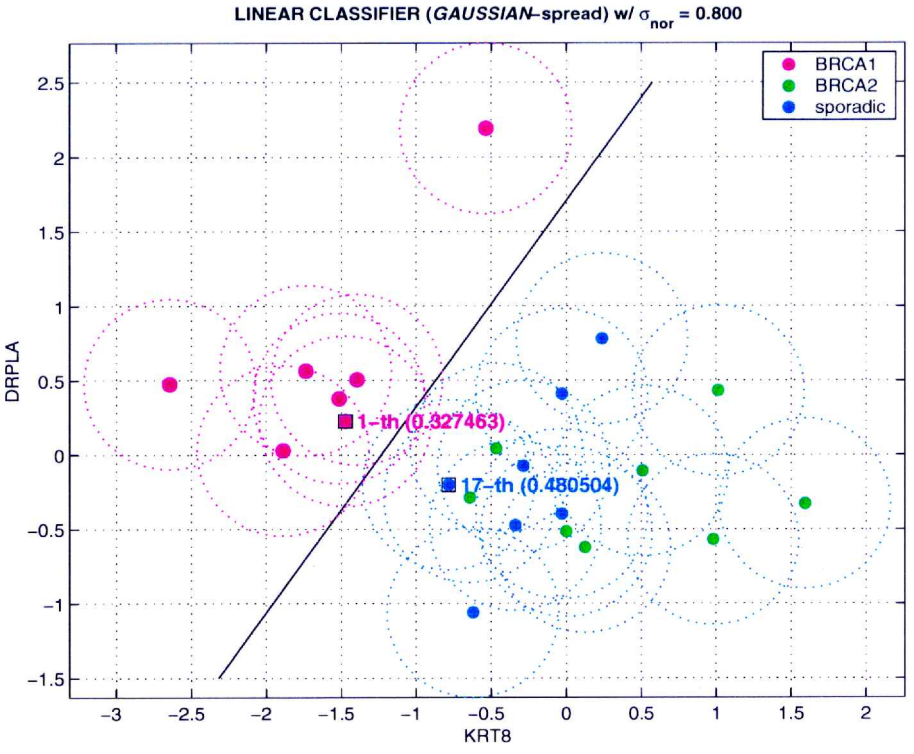


FIG. 4. Linear classifier constructed with a spreading, $\sigma_{\text{nor}} = 0.8$.

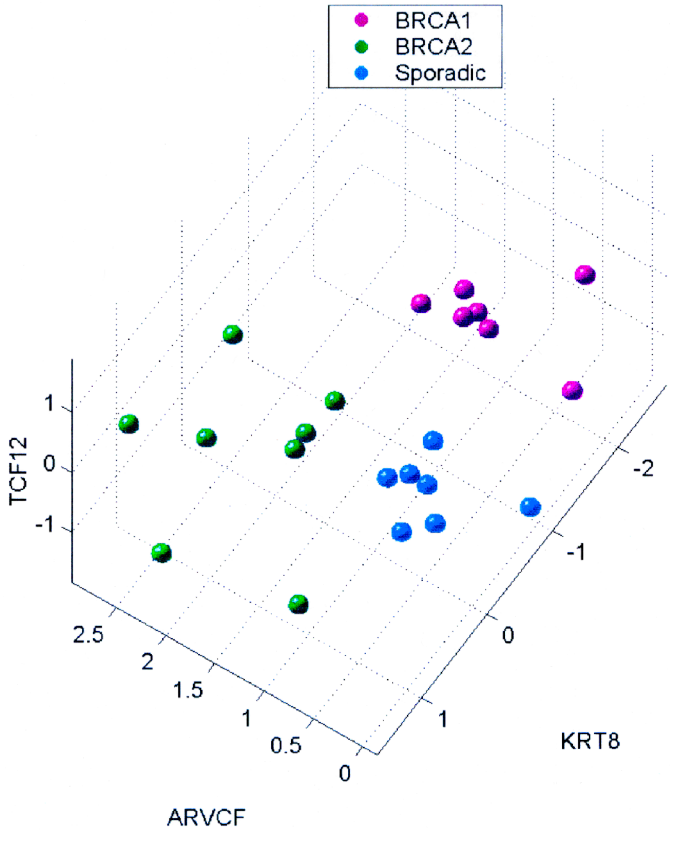


FIG. 5. Separation of three tumor classes.

Table 2. Feature Sets to Classify BRCA2 Tissues Versus BRCA1 and Sporadic Tissues

	Feature set			Error (%) $\sigma_{nor} = 0.6$	Error (%) $\sigma_{nor} = 0.8$
Two genes	ARVCF	PHYH		6.1	9.8
	ZNF161	COX6C		8.4	12.4
	ARVCF	TCF12		9.7	13.5
	ZNF161	TSC2		10.1	13.3
	FGFR3	PDCD5		10.1	13.7
	ARHC	DCTN4		10.9	15.5
	ZNF161	PRO1659		11.6	15.6
	SUPT4H1	EST		12.0	16.8
	EST(358333)	RPS6KB1		12.0	16.8
	RBL2	SUPT4H1		12.5	17.5
	MCAM	CLTC		13.8	18.8
	MCAM	SUPT4H1		15.0	19.7
	ARHC	EST(137417)		18.2	22.1
Three genes	UGTREL1	GNA12	CDK4	2.5	4.8
	GDI2	MTMR4	CDK4	2.7	5.1
	NOP56	UGTREL1	PCNA	3.2	5.9
	FGFR3	VDR	PPP1CB	2.8	6.1
	SLC2A5	UGTREL1	CDK4	3.2	6.2
	MAPK1	ACTR1A	PCNA	3.4	6.5
	GDI2	ESTs	PCNA	4.0	6.7
	ARVCF	D123	ITGB2	3.4	6.7
	PPP1CB	ARVCF	VAV3	3.8	7.0
	SERPINE1	UGTREL1	PPP1CB	4.0	7.7

A practical issue concerns which individual genes are useful for classification. Tables 3 and 4 show genes that appear most often in the lists of strong performing gene sets for d equal to 2 through 5, along with the number of lists in which the gene appears.

When developing a gene expression-based classifier for a heavily studied system, one expects that some of the genes identified will have been previously noted as being differentially expressed in that system. When the method used to develop the classifiers finds genes that are most differentially expressed between the classes, one further expects that these genes may have been seen in other contexts. This is based on the tendency of genes with highly variable expression to exhibit this variability in a variety of circumstances and tissues. Both of these expectations are clearly met in this case.

Some of the strong classifiers separating BRCA1 from BRCA2 and sporadic cancers are quite familiar. One of the strongest classifiers, keratin 8 (KRT8), is a member of the cytokeratin family of genes. Cytokeratins are frequently used to identify breast cancer metastases by immunohistochemistry, and cytokeratin 8 abundance has been shown to correlate well with node-positive disease (Brotherick *et al*., 1998). Another top classifier is the cyclin D1 gene (PRAD1). Although discovered by virtue of its association with parathyroid adenomas, this gene has long been associated with breast cancer (Bartkova *et al*., 1994; Wang *et al*., 1994). Similarly, the tumor-associated antigen L-6 (TM4SF1), is a member of a family of integral membrane proteins, several of which are also overexpressed in tumors (Marken *et al*., 1992). Antigen L-6 is frequently over-expressed in carcinomas, and antibody binding to L-6 on tumors in nude mouse models inhibits their outgrowth (Hellstrom *et al*., 1986). Receptors and genes interacting with them are well represented among the strong classifiers. An unusual version of the GABA receptor, gamma-aminobutyric acid A receptor pi (GABRP), which has been shown to alter the sensitivity of GABA receptors to the steroid pregnanolone in the uterus (Hedblom *et al*., 1997) is found to be more highly expressed in BRCA1 tumors, which are known for their lack of ER and PR receptors. The gene TOB1, which interacts with the oncogene receptor ERBB2, is found to be more highly expressed in BRCA2 and sporadic cancers, which are likewise more likely to harbor ERBB2 gene amplifications. TOB1 has an antiproliferative activity that is apparently antagonized by ERBB2 (Matsuda *et al*., 1996).

Table 3. Strong Genes Found for Classifying BRCA1 Versus BRCA2 and Sporadic

Clone ID	*#*	*Gene descriptions*
897781	41	keratin 8
375635	17	transcription factor 12 (HTF4, helix-loop-helix transcription factors 4)
45291	11	dentatorubral-pallidoluysian atrophy (atrophin-1)
307843	9	eukaryotic translation initiation factor 2C, 2
823940	7	Tob1, transducer of ERBB2, 1
950682	5	phosphofructokinase, platelet
247818	5	Homo sapiens cDNA FLJ13495 fis, clone PLACE1004425
46182	5	CTP synthase
32790	3	mutS (E. coli) homolog 2 (colon cancer, nonpolyposis type 1)
840567	3	L6 antigen, transmembrane 4 superfamily member 1
280768	2	L6 antigen, transmembrane 4 superfamily member 1
40959	2	solute carrier family 25 (mitochondrial carrier; adenine nucleotide translocator), member 5
233194	2	special AT-rich sequence binding protein 1 (binds to nuclear matrix/scaffold-associating DNA's)
66977	2	androgen induced protein
525518	2	ubiquitin specific protease 7 (herpes virus-associated)
841641	2	cyclin D1 (PRAD1: parathyroid adenomatosis 1)
144926	2	ESTs, Weakly similar to B0495.6 [C.elegans]
47884	2	macrophage migration inhibitory factor (glycosylation-inhibiting factor)
52650	2	v-ets avian erythroblastosis virus E26 oncogene homolog 2
366647	2	butyrate response factor 1 (EGF-response factor 1)

There are two further observations derived from the development of these classifiers that are in keeping with some of the expression patterns observed among these cancers. One pattern of expression noted in the BRCA1 tumors was an increase in expression of genes involved in DNA repair and apoptosis, indicating a chronic stress response. One of the strong classifiers found was crystallin alpha B (CRYAB), a member of the small heat shock protein family relatively over-expressed in BRCA1 tumors. High levels

Table 4. Genes Found for Classifying BRCA1 Versus BRCA2

Clone ID	*#*	*Gene descriptions*
897781	30	keratin 8
950682	25	phosphofructokinase, platelet
839736	21	crystallin, alpha B
841641	18	cyclin D1 (PRAD1: parathyroid adenomatosis 1)
82991	14	ectonucleotide pyrophosphatase/phosphodiesterase 1
563598	7	gamma-aminobutyric acid (GABA) A receptor, pi
814270	5	polymyositis/scleroderma autoantigen 1 (75kD)
564803	4	forkhead box M1
214068	4	GATA-binding protein 3
823940	4	Tob1, transducer of ERBB2, 1
26184	3	phosphofructokinase, platelet
810551	3	low density lipoprotein-related protein 1 (alpha-2-macroglobulin receptor)
783729	3	ERBB2, v-erb-b2 avian erythroblastic leukemia viral oncogene homolog 2 (neuro/glioblastoma derived oncogene homolog)
33794	3	ATP synthase, H+ transporting, mitochondrial F1 complex, gamma polypeptide 1
211758	2	Homo sapiens cDNA: FLJ22256 fis, clone HRC02860
752631	2	fibroblast growth factor receptor 3 (achondroplasia, thanatophoric dwarfism)
711826	2	KIAA0019 gene product
135118	2	GATA-binding protein 3
840567	2	L6 antigen, transmembrane 4 superfamily member 1
245422	2	adducin 1 (alpha)

Table 5. Strong Genes Found for Classifying BRCA1 + BRCA-Like Sporadic Versus BRCA2 and Sporadic

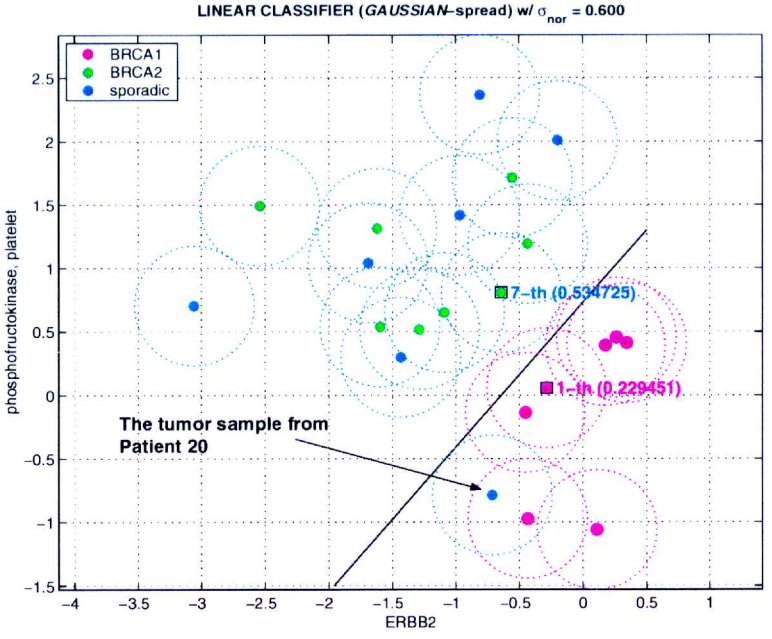
Clone ID	*#*	*Gene descriptions*
897781	18	keratin 8
823940	17	Tob1, transducer of ERBB2, 1
563598	12	gamma-aminobutyric acid (GABA) A receptor, pi
841641	12	cyclin D1 (PRAD1: parathyroid adenomatosis 1)
280768	10	L6 antigen, transmembrane 4 superfamily member 1
840567	7	transmembrane 4 superfamily member 1
725454	4	CDC28 protein kinase 2
768370	4	tissue inhibitor of metalloproteinase 3 (Sorsby fundus dystrophy, pseudoinflammatory)
564803	4	forkhead box M1
81331	4	NA128†
365147	4	ERBB2, v-erb-b2 avian erythroblastic leukemia viral oncogene homolog 2 (neuro/glioblastoma derived oncogene homolog)
42888	4	interleukin enhancer binding factor 2, 45kD
839736	3	crystallin, alpha B
290871	3	Integrin alpha-3 subunit
141768	3	ERBB2, v-erb-b2 avian erythroblastic leukemia viral oncogene homolog 2 (neuro/glioblastoma derived oncogene homolog)
814595	3	protein kinase C binding protein 1
359119	3	CDC28 protein kinase 2
49888	2	ADP-ribosylation factor 4-like
825470	2	topoisomerase (DNA) II alpha (170kD)
133236	2	ESTs

† NA128 - not assigned to clusters in UniGene Build 128

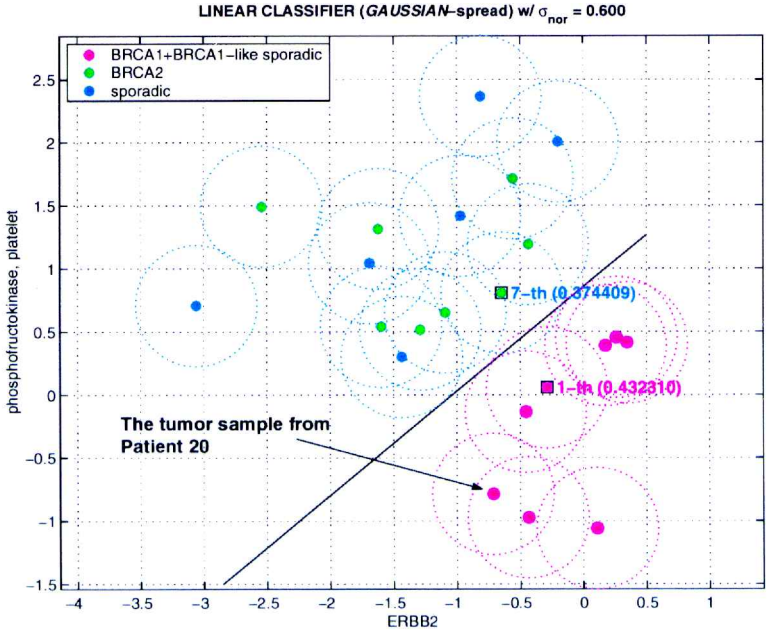
of expression of this protein are associated with stress and can enhance cells' resistance to stress induced by DNA damage (Andley *et al.*, 1998). A final noteworthy result further illustrates the sensitivity of the method. Standard statistical methods used in the published study (Hedenfalk *et al.*, 2001) to find genes with classifying power identified the ERBB2 (LO3E9 or HV54A3) gene as a classifier, and it could be seen that this gene was relatively underexpressed in BRCA1 genes, though there was one significant outlier case among the sporadic tumors. Gene ERBB2 was not found as a strong classifier using the evaluation reported in this chapter. The analysis of the expression profiles of the full set of tumors suggested that the ERBB2 outlier, sporadic cancer (the tumor sample from Patient 20), had an expression pattern unusually similar to a BRCA1 tumor as shown in Fig. 6. Further examination of this tumor indicated that there was substantial methylation of the promoter region of the BRCA1 gene in this individual, making it possible that this individual had sufficiently low BRCA1 expression to mimic the effects of a gene mutation. When a classifier for BRCA1 versus BRCA2 and sporadic cancers was built with the BRCA1-like sporadic case classed with the authentic BRCA1 mutants, the ERBB2 gene was once again identified as a strong classifier as shown in Table 5. This demonstrates that this procedure exhibits the expected strong exclusion of genes that have even a single outlier.

7. CONCLUSION

The proposed parameterized classifier-design algorithm yields classifiers whose strength is measured in terms of the variance of a distribution derived by spreading the sample data. In the case of linear classifiers, the classifier can be derived by classical Wiener-filter theory. This avoids time-consuming stochastic design methodologies and is critical for genomic applications in which the set of potential features can be very large. The algorithm has been applied to find features that can discriminate between BRCA1 and BRCA2 breast cancers, and it has performed in a more sensitive fashion than previous methods to achieve such discrimination. The strategy employed allows one to find small numbers of genes having the greatest



(a) BRCA1 versus BRCA2 and sporadic



(b) BRCA1 and BRCA-like sporadic versus BRCA2 and sporadic

FIG. 6. The tumor sample from patient 20 as outlier in ERBB2 expression.

discriminating power, thereby allowing researchers to quickly focus on the most promising candidates for diagnostics and therapeutics. Owing to the need to search among many feature sets, practical application of the algorithm is suboptimal; however, intelligent searching has resulted in well-performing feature sets. In the present paper, full searches have been implemented for one and two features, and ten million feature sets have been examined within the search procedure for larger feature sets. With the completion of the new 5,000 CPU machine, much greater numbers of feature sets will be tested.

A. APPENDIX: ANALYTICAL ERROR REPRESENTATION

We expand Equation 14 for the analytic error representation. For the uniform circular distribution with radius R, consider a sample data point $\mathbf{x}_k$ lying in the correct region. The contribution of $\mathbf{x}_k$ to the distribution of the sample points after spreading by Z is depicted in Fig. 7. For $0 < h < R$, the corresponding contribution to the misclassification error (prior to normalization) is given by

$$\varepsilon_d(h, R) = \int_h^R V_{d-1}\left(\sqrt{R^2 - x_j^2}\right) dx_j \tag{24}$$

where $V_d(R)$ is the hypervolume of the sphere of radius R in $\Re^d$. A change of variables yields

$$\begin{aligned}\varepsilon_d(h, R) &= R^d V_{d-1} \int_{h/R}^1 \left(1 - y_j^2\right)^{\frac{d-1}{2}} dy_j \\ &= R^d \varepsilon_d(h/R)\end{aligned} \tag{25}$$

where $\varepsilon_d(h/R)$ is the error for the unit hypersphere and V_{d-1} is the hypervolume of the unit hypersphere in $\Re^d$. For $h \geq R$, $\varepsilon_d(h, R) = 0$. If $\mathbf{x}_k$ lies on the wrong side of the hyperplane, then its contribution to the error after spreading will be $V_d(R) - \varepsilon_d(h, R)$. With arbitrary d, the integral is problematic (in fact, this can be represented using *hypergeometric* function), but it is not difficult for fixed d. For example,

$$\begin{aligned}\varepsilon_1(h, R) &= R - h = R\left(1 - \frac{h}{R}\right) = R\varepsilon_1\left(h^*\right) \\ \varepsilon_2(h, R) &= \frac{\pi R^2}{2} - h\sqrt{R^2 - h^2} - \arcsin\left(\frac{h}{R}\right) R^2 \\ &= R^2\left(\frac{\pi}{2} - h^*\sqrt{1 - h^{*2}} - \arcsin\left(h^*\right)\right) = R^2\varepsilon_2\left(h^*\right) \\ \varepsilon_3(h, R) &= \frac{2\pi R^3}{3} - \pi R^2 h + \frac{\pi h^3}{3} \\ &= R^3\left(\frac{2\pi}{3} - \pi h^* + \frac{\pi}{3}h^*\right) = R^3\varepsilon_3\left(h^*\right)\end{aligned} \tag{26}$$

where $h^* = h/R$.

To obtain a normalized total error for the sample $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, let $\mathrm{T}(\mathbf{x}_k) = 0$ if $\mathbf{x}_k$ is classified correctly and $\mathrm{T}(\mathbf{x}_k) = 1$ otherwise. Normalizing by the volume of the n spheres for the sample yields the misclassification error

$$\begin{aligned}\varepsilon_\sigma &= \frac{1}{nV_d(R)} \sum_{k=1}^n \left[\mathrm{T}(\mathbf{x}_k)\left(V_d(R) - \varepsilon_{k,d}(h_k, R)\right) + (1 - \mathrm{T}(\mathbf{x}_k))\,\varepsilon_{k,d}(h_k, R)\right] \\ &= \frac{R^d}{nV_d(R)} \sum_{k=1}^n \left[\mathrm{T}(\mathbf{x}_k)\left(V_d - \varepsilon_{k,d}\left(h_k^*\right)\right) + (1 - \mathrm{T}(\mathbf{x}_k))\,\varepsilon_{k,d}\left(h_k^*\right)\right]\end{aligned} \tag{27}$$

where h_k is the distance from x_k to the hyperplane, $h_k^* = h_k/R$, and V_d is the hypervolume of the unit hypersphere in $\Re^d$.

For Gaussian distribution, the error contribution for each sample point is given in terms of the error function by $\varepsilon_d(h, \sigma) = (1 - \mathrm{erf}(h/\sqrt{2}\sigma))/2$, which includes the normalization as part of the error function. The quantity ε_σ is found by summing the contributions and then dividing by n.

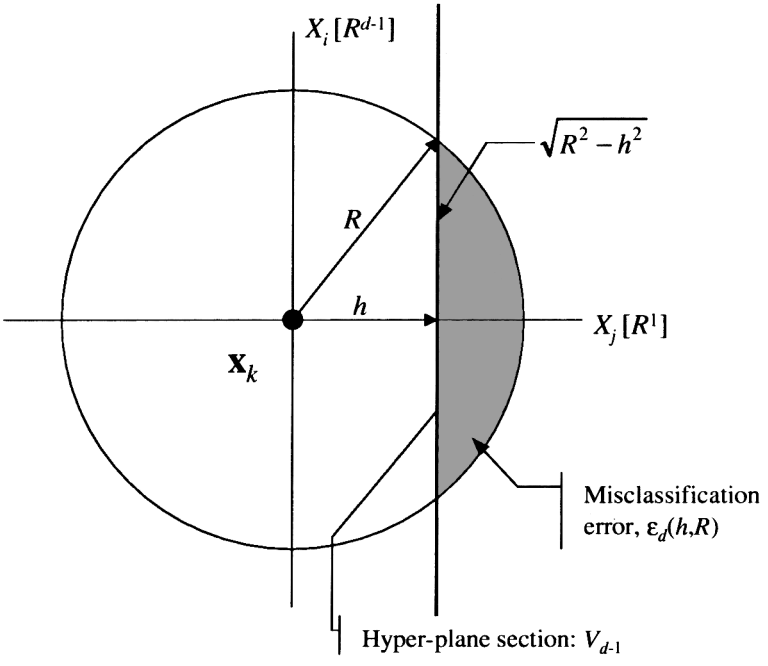


FIG. 7. Misclassification error for a sample data with spreading.

8. ACKNOWLEDGMENT

We wish to thank Drs. Edward B. Suh and Robert L. Martino for providing the computational resource of Beowulf clustered supercomputer at the Center for Information Technology of NIH for the heavy computation of the algorithm.

REFERENCES

Ali, M.M., and Storey, C. 1994. Modified controlled random search algorithms. *Int. J. Comp. Math.* 53, 229–235.

Alizadeh, A. *et al*. 1999. The lymphochip: A specialized cDNA microarray for the genomic-scale analysis of gene expression in normal and malignant lymphocytes. *Cold Spring Harbor Symp. Quant. Biol.* 64, 71–78.

Alon, U. *et al*. 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA* 96, 6745–6750.

Andley, U.P. *et al*. 1998. The molecular chaperone alphaA-crystallin enhances lens epithelial cell growth and resistance to UVA stress. *J. Biol. Chem.* 273(47), 31252–31261.

Bartkova, J. *et al*. 1994. Cyclin d1 protein expression and function in human breast cancer. *Int. J. Cancer* 57(3), 353–361.

Baum, E. 2000. On the capabilities of multilayer perceptrons. *Complexity* 4, 193–215.

Ben-Dor, A. *et al*. 2000. Tissue classification with gene expression profiles. *J. Comp. Biol.* 7, 559–583.

Bittner, M. *et al*. 2000. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature* 406(6795), 536–540.

Bresina, J.L. 1996. Heuristic-biased stochastic scheduling. In *Proc. Natl. Conf. Artif. Intell. (AAAI96)*, 271–278, Cambridge, MA, AAAI Press.

Brotherick, I. *et al*. 1998. Cytokeratin expression in breast cancer: Phenotypic changes associated with disease progression. *Cytometry* 32, 301–308.

Cybenko, G. 1989. Approximation by superposition of sigmoidal functions. *Math. Control, Signals, Systems* 2, 303–314.

DeRisi, J.L. *et al*. 1997. Exploring metabolic and genetic control of gene expression on a genomic scale. *Science* 278, 680–686.

Devroye, L. *et al*. 1996. *A Probabilistic Theory of Pattern Recognition*, Springer, New York.

Dougherty, E.R. 1999. *Random Processes for Image and Signal Processing. Imaging Science and Engineering*, SPIE Press and IEEE Press, Bellingham, WA.

Dougherty, E.R. 2001. Small sample issues for microarray-based classification. *Comp. Funct. Genomics* 2, 28–34.

Duggan, D.J. *et al*. 1999. Expression profiling using cdna microarrays. *Nature Genet*. 21, 10–14.

Funahashi, K. 1989. On the approximate realization of continuous mappings by neural networks. *Neural Networks* 2, 183–192.

Goldberg, D.E. 1989. *Genetic Algorithms: In Search, Optimizations and Machine Learning*, Addison Wesley, New York.

Golub, T.R. *et al*. 1999. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* 286, 531–537.

Hedblom, E., and Kirkness, E.F. 1997. A novel class of GABAA receptor subunit in tissues of the reproductive system. *J. Biol. Chem.* 272(24), 15346–15350.

Hedenfalk, I. *et al*. 2001. Gene expression profiles in hereditary breast cancer. *New Engl. J. Med.* 344, 539–548.

Hellstrom, I. *et al*. 1986. Antitumor effects of l6, an IgG2a antibody that reacts with most human carcinomas. *Proc. Natl. Acad. Sci. USA* 83, 7059–7063.

Hogg, T. 1996. Quantum computing and phase transitions in combinatorial search. *J. Artif. Intell. Res.* 4, 91–128.

Hornik, K. *et al*. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks* 2, 359–366.

Kanoh, H. *et al*. 1995. Genetic algorithms for constraint satisfaction problems. In *Proc. IEEE Int. Conf. System, Man, Cybernetics*, 626–631, Piscataway, NJ, IEEE Press.

Khan, J. *et al*. 1998. Gene expression profiling of alveolar rhabdomyosarcoma with cDNA microarrays. *Cancer Res.* 58(22), 5009–5013.

Khan, J. *et al*. 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nat. Med.* 7, 673–679.

Li, J., and Rhinehart, R.R. 1998. Heuristic random optimization. *Comput. Chem. Eng*. 22, 427–444.

Marken, J.S. *et al*. 1992. Cloning and expression of the tumor-associated antigen 16. *Proc. Natl. Acad. Sci. USA* 89, 3503–3507.

Masri, S.F. *et al*. 1980. Global optimization algorithm using adaptive random search. *Appl. Math. Comput*. 7, 353–376.

Matsuda, S. *et al*. 1996. Tob, a novel protein that interacts with p185erbB2, is associated with anti-proliferative activity. *Oncogene* 12, 705–713.

Mohan, C., and Nguyen, H.T. 1999. Controlled random search technique incorporating the simulated annealing concept for solving integer and mixed integer global optimization problems. *Comput. Optim. Appl*. 14, 103–132.

Perou, C.M. *et al*. 2000. Molecular portraits of human breast tumours. *Nature* 406, 747–752.

Price, W.L. 1983. Global optimization by controlled random search. *J. Optimiz. Theory Appl*. 40, 333–348.

Rubinstein, Y.R., and Samorodnitsky, G. 1982. Efficiency on the random search method. *Math. Comp. Simul*. 24, 257–268.

Schena, M. *et al*. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270, 467–470.

Solis, F.J., and Wets, R.J.-B. 1981. Minimization by random search techniques. *Math. Oper. Res.* 6, 19–30.

Srinivas, M., and Patnaik, L.M. 1994. Genetic algorithms: A survey. *Computer* 27, 17–26.

Ustyuzhaninov, V.G. 1980. Effectiveness of random search. *Autom. Contr. Comp. Sci.* 14, 65–67.

Vapnik, V., and Chervonenkis, A. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory Prob. Appl*. 16, 264–280.

Wang, T.C. *et al*. 1994. Mammary hyperplasia and carcinoma in MMTV-cyclin d1 transgenic mice. *Nature* 369(6482), 669–671.

Address correspondence to:
Edward R. Dougherty
Texas A&M University
Department of Electrical Engineering
3128 TAMU
College Station, TX 77843-3128

E-mail: edward@ee.tamu.edu