Pattern Recognition Letters 29 (2008) 1667–1674

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

The peaking phenomenon in the presence of feature-selection

Chao Sima [a,*], Edward R. Dougherty [a,b]

[a] Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ 85004, USA
[b] Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, USA

ARTICLE INFO

Article history:
Received 18 September 2007
Received in revised form 20 February 2008
Available online 4 May 2008

Communicated by L. Heutte

Keywords:
Classification
Feature-selection
Peaking phenomenon

ABSTRACT

For a fixed sample size, a common phenomenon is that the error of a designed classifier decreases and then increases as the number of features grows. This peaking phenomenon has been recognized for forty years and depends on the classification rule and feature-label distribution. Historically, the peaking phenomenon has been treated by assuming a fixed ordering of the features, usually beginning with the strongest individual feature and proceeding with features of decreasing individual classification capability. This does not take into account feature-selection, which is commonplace in high-dimensional and small sample settings. This paper revisits the peaking phenomenon in the presence of feature-selection. Using massive simulation in a high-performance computing environment, the paper considers various combinations of feature-label models, feature-selection algorithms, and classifier models to produce a large library of error versus feature size curves. Owing to the prevalence of feature-selection in genomic classification, we also consider gene-expression-based classification of breast-cancer patient prognosis. Results vary widely and are strongly dependent on the combination. The error curves tend to fall into three categories: peaking, settling into a plateau, or falling very slowly over a long range of feature set sizes. It can be concluded that one should be wary of applying peaking results found in the absence of feature-selection to settings in which feature-selection is employed.

1. Introduction

High-throughput technologies such as gene and protein expression microarrays offer the ability to simultaneously measure vast numbers of biological variables. This has given rise, both in genomics and proteomics, to the hope that these measurements can be used as classifier features for medical diagnosis. The difficulty is that the classical relationship between the number of features and the number of data points in the sample is typically reversed. Rather than there being a large number of data points and a small number of potential features, sample sizes tend to be small and there are thousands of potential features. This situation manifests itself in the peaking phenomenon: employing too large a number of features yields poorer classification accuracy than using a small number of features. The peaking phenomenon was first rigorously demonstrated for discrete classification (Hughes, 1968), but it has wide-ranging effects, depending on the classification rule and feature-label distribution (Hua et al., 2005a,b; Jain and Waller, 1978; Navot et al., 2006; Raudys, 1979; Raudys and Jain, 1991; Trunk, 1979). The potential downside of using too many features is most critical for small samples, which are commonplace in high-throughput genomic and proteomic applications. The peaking phenomenon leads to the need for feature-selection based on the sample data. This paper examines the behavior of the peaking phenomenon in relation to feature-selection.

A key issue concerning feature-selection is error monotonicity, or the lack of it. Given a feature set and full knowledge of the feature-label distribution, the Bayes error, which is the error of an optimal classifier for a feature set, is monotone: if A and B are feature sets for which $A \subset B$, then $\varepsilon_B \leqslant \varepsilon_A$, where ε_A and ε_B are the Bayes errors corresponding to A and B, respectively. However, if $\varepsilon_{A,n}$ and $\varepsilon_{B,n}$ are the corresponding errors resulting from designed classifiers on a sample of size n, then it cannot be asserted that $\varepsilon_{A,n} \geqslant \varepsilon_{B,n}$. It may even be that $E[\varepsilon_{B,n}] > E[\varepsilon_{A,n}]$, the expected error for the larger feature set is larger than that for the smaller features set. This is what leads to the peaking phenomenon, which in its simplest manifestation takes the form of decreasing expected error following by increasing expected error for increasingly large feature sets. Given a sequence, $x_1, x_2, \ldots, x_d, \ldots$, of features, at first there is decrease in expected error as d increases and then, after some point, an increase in error for increasing d.

To illustrate the classical idea of peaking, we consider an example in which the class-conditional distributions are Gaussian with identical covariance matrix $\mathbf{K}$, the classes are equally likely, and the Bayes classifier results from linear discriminant analysis (LDA) (Hua et al., 2005b). There are 30 available features (and

* Corresponding author. Tel.: +1 602 343 8485; fax: +1 602 343 8740.
E-mail address: csima@tgen.org (C. Sima).
URL: http://compbio.tgen.org/paper_supp/fs_peaking (C. Sima).

0167-8655/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2008.04.010

hence the peaking phenomenon will only show up in the graphs for which peaking occurs with less than 30 features). We assume the basic blocked covariance structure

$$\mathbf{K} = \sigma^2 \underbrace{\begin{bmatrix} \Sigma_\rho & 0 & \cdots & 0 \\ 0 & \Sigma_\rho & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_\rho \end{bmatrix}}_{G \text{ groups}} \tag{1.1}$$

where

$$\Sigma_\rho = \begin{bmatrix} 1 & \rho & \ldots & \rho \\ \rho & 1 & \ldots & \rho \\ \vdots & \vdots & \ddots & \vdots \\ \rho & \rho & \ldots & 1 \end{bmatrix} \tag{1.2}$$

Features within the same block are correlated with correlation coefficient ρ and features within different blocks are uncorrelated. There are G groups, with G being a divisor of 30, and $r = 30/G$ is the number of features in each group. We denote a particular feature with the label x_{ij}, where i, $1 \leqslant i \leqslant G$, denotes the group to which the feature belongs and j, $1 \leqslant j \leqslant r$, denotes its position in the group. We list the features in the order $x_{11}, x_{21}, \ldots, x_{G1}, x_{12}, \ldots, x_{Gr}$. Fig. 1 shows the effect of correlation on peaking. Note that the sample size must exceed the number of features to avoid degeneracy. The variance σ^2 is set to give a Bayes error of 0.05. In part (a) of the figure, there are $G = 5$ groups and $\rho = 0.125$. Peaking occurs with very few features for sample sizes 30 and below, but then exceeds 30 features for sample sizes above 90. Matters are different in part (b), where there is a single group and the features are highly correlated, $\rho = 0.5$. Here, even with a sample size of 200, the optimal number of features is only 8. The behavior in Fig. 1 corresponds to the usual understanding of the peaking phenomenon; however, the situation can be far more complicated (Hua et al., 2005b).

The main issue for the present paper is that, heretofore, as in the example of Fig. 1, the peaking phenomenon has been studied without feature-selection: a model is assumed in which the order of the features is given and the issue is to find the number of features that yields the lowest classification error. For this approach to achieve the best results, we would have to know the optimal feature set for each dimension (number of features). In applications, this is not known. In practice, we are confronted by a fundamental limiting principle: to select a subset of k features from a set of d features and be assured that it provides an optimal classifier with minimum error among all optimal classifiers for subsets of size k, all k-element subsets must be checked unless there is distributional knowledge that mitigates the search requirement. This principle is formalized in the following theorem (Cover and Van Campenhout, 1977), in which $\varepsilon[A]$ denotes the Bayes error corresponding to feature set A: if $\{U_1, U_2, \ldots, U_r\}$ is the family of all possible feature sets formed from the set $\{X_1, X_2, \ldots, X_k\}$ of random variables under the assumption that, $i < j$ if $U_i \subset U_j$, then there exists a distribution of the random variables $X_1, X_2, \ldots, X_k, Y$, where Y is binary, such that $\varepsilon[U_1] > \varepsilon[U_2] > \cdots > \varepsilon[U_r]$. The requirement that $i < j$ if $U_i \subset U_j$ is necessary because the subset condition implies that $\varepsilon[U_i] \geqslant \varepsilon[U_j]$. To avoid an exhaustive search, which is impossible except when there is a very small number of available features, many suboptimal algorithms have been proposed for feature-selection. The purpose of the present paper is to examine the peaking phenomenon in high-dimensional settings when feature-selection is required, and to do so in the kind of small sample setting typical in genomic applications. Small samples tend to hurt a feature-selection algorithm because they adversely impact the estimation of parameters used by the algorithm (Sima et al., 2005, Zhou and Mao, 2006).

When there is feature-selection, the feature-selection algorithm is part of the classification rule used to design the classifier. Feature-selection may occur integrally with the classification rule, the *wrapper* method, or it may occur prior to the application of a standard classification rule applied to a subset of features, the *filter* method, or it may be a combination of both. In any case, feature-selection is part of the overall classification rule and, relative to this rule the number of variables is the number in the data measurements, not the final number used in the designed classifier. Feature-selection results in a subfamily of the original family of classifiers, and thereby constitutes a form of constraint. For instance, if there are D features available and LDA is used directly, then the classifier family consists of all hyperplanes in D-dimensional space, but if a feature-selection algorithm reduces the number of variables to $d < D$ prior to application of LDA, then the classifier family consists of all hyperplanes in D-dimensional space that are confined to d-dimensional subspaces. Feature selection yields classifier constraint, not a reduction in the dimensionality of the feature space relative to design. Because we want to mix different feature-selection procedures with different classifier functions, we will refer to the latter as "classifier rules", so that the full classification rule consists of a feature-selection algorithm and a classifier rule.

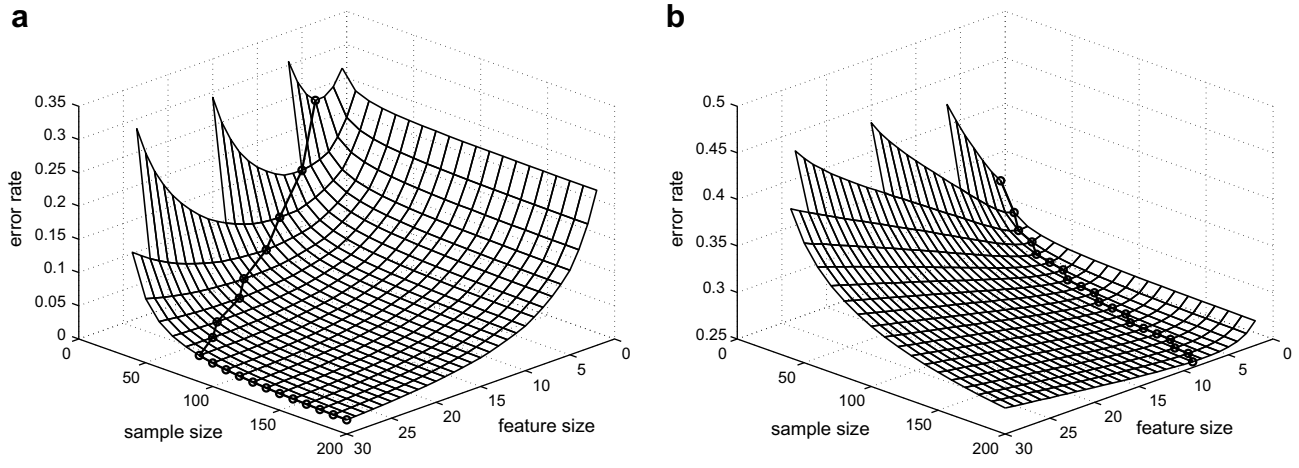

Fig. 1. Optimal number of features for LDA in linear model: (a) slightly correlated features and (b) highly correlated features.

If one uses cross-validation to estimate the error of a designed classier, then the cross-validation data splitting must be done outside the classification rule to maintain its approximate unbiasedness, and this means that feature-selection is done separately for each cross-validation data split. Over the years, a number of papers have investigated the effects of estimating classifier error directly on the training data (Murray, 1977, Ambroise and McLachlan, 2002, Singhi and Liu, 2006, Li et al., 2007). The problem is clear: the features will be selected so as to best discriminate on the training data, and therefore feature-selection can increase overfitting the data in classifier design.

Using feature-selection when studying the peaking phenomenon requires high-performance computing capability, and this has certainly been a limiting factor heretofore. We utilize a Beowulf cluster to run many simulations and build up the large set of results posted on the companion website. We will use various feature-label models, feature-selection algorithms, and classifier models.

The key conclusion will be that one must be very careful in making any type of general statements about peaking when feature-selection is involved, except perhaps that there are no general statements; nonetheless, we will see that there are three common trends for the error curve as a function of the number of features. While we believe that a large study such as this one is beneficial in its own right because it elucidates the various kinds of peaking behavior one observes across a wide range models, especially insofar as those models correspond to the kinds of feature interactions one might expect in genomic or proteomic data, the paper also provides a cautionary warning to those involved in high-dimensional feature-selection: before applying a feature-selection algorithm in a particular setting, do a preliminary simulation study to examine the peaking effects of a proposed feature-selection algorithm in conjunction with the proposed classifier rule, using a model that is as close to the data as possible, for instance, by fitting a model to the data. While this approach cannot provide guarantees, it can at least provide one with warnings.

2. Method

The simulation problem is to find average errors for classifiers designed from sample data using different feature sizes d, various classifier rules $\mathscr{R}$, and feature-selection methods $\mathscr{F}$. To do this, we utilize the following procedure:

1. Generate a sample set S of size n and a total of D features from a feature-label model $\mathscr{M}$.
2. Select a size-d feature set A using a feature-selection method $\mathscr{F}$ on S. The "best" features are derived from the data model $\mathscr{M}$ directly.
3. Design a classifier ψ from S for the feature set A according to the classifier rule $\mathscr{R}$.
4. Compute the error ε_d for ψ using the underlying distribution of the model.
5. Repeat steps 2 through 4 for $d \in \{1, 2, \ldots, d_{\max}\}$.
6. Repeat steps 1 through 5 N times and compute $\bar{\varepsilon}_d$, the average of the ε_d's.
7. Repeat steps 1 through 6 for different models $\mathscr{M}$, different feature-selection methods $\mathscr{F}$, and different classifier rules $\mathscr{R}$.

2.1. Data model

We consider two types of data models. The first type, which is the *basic*, or *noise-free* model, includes the following 10 Gaussian models, labeled $M1$ through $M10$. Each is a two-class Gaussian model with equally likely classes and class-conditional densities having covariance matrices Σ_1 and Σ_2. One class mean is located at the origin $\vec{0}$ and the other at $\vec{\mu}$, with the location of $\vec{\mu}$ depending on the model.

- M1: A simple linear model in which $\Sigma_1 = \Sigma_2 = \mathbf{I}$, the identity matrix, so that all features are uncorrelated. Let $\vec{\mu} = [a_1 a_2 \ldots a_D]$ and G be the total number of groups we divide the D features into. We set up $\vec{\mu}$ so that $a_i = \delta \times (1 - \lfloor \frac{i-1}{D/G} \rfloor / G)$, where δ is a prescribed constant and $\lfloor \cdot \rfloor$ is the floor function. For $i < j$, $a_i \geqslant a_j$.
- M2: Similar to $M1$ but with $\Sigma_1 = \mathbf{I}$ and $\Sigma_2 = c\mathbf{I}$, where c is a constant and $c \neq 1$. The Bayesian decision boundary is quadratic.
- M3: The correlation between any two features is the same, so we have $\Sigma_1 = \Sigma_2 = \Sigma_\rho$ and Σ_ρ has the same structure as in Eq. (1.2). $\vec{\mu}$ is set up the same way as in $M1$.
- M4: Similar to $M3$, but with $\Sigma_1 = \Sigma_\rho$ and $\Sigma_2 = c\Sigma_\rho$.
- M5: This is a *Block Covariance Model* where all features are equally divided into G groups. The features from different groups are uncorrelated and the features from the same group possess the same correlation, ρ, among each other. The structure of the covariance matrix is

$$\Sigma_1 = \Sigma_2 = \Sigma_B = \underbrace{\begin{bmatrix} \Sigma_\rho & 0 & \cdots & 0 \\ 0 & \Sigma_\rho & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \Sigma_\rho \end{bmatrix}}_{G \text{ blocks}} \tag{2.1}$$

  $\vec{\mu}$ is different from $M1$ as it is fixed at $[\delta\delta \cdots \delta]$.
- M6: Similar to $M5$, but with $\Sigma_1 = \Sigma_B$ and $\Sigma_2 = c\Sigma_B$.
- M7: This is a *Bimodal Mixture Model* that consists of two equally likely distributions, with one of them being a simple Gaussian with mean at $\vec{0}$ and covariance matrix $\Sigma_1 = \mathbf{I}$, and the other being a mixture (with equal probabilities) of two Gaussians, with means at $\mu^{[1]}$ and $\mu^{[2]} = -\mu^{[1]}$, and covariance matrix for both components of the mixture being $\Sigma_2 = \Sigma_1 = \mathbf{I}$. $\mu^{[1]}$ is set up the same way as in $M1$.
- M8: Similar to $M7$, but with $\Sigma_1 = \mathbf{I}$ and $\Sigma_2 = c\mathbf{I}$.
- M9: Similar to $M7$, but with different locations for the means of the mixture Gaussian part. Specifically, for $i = 1, 2, \cdots, D$, we let $\mu_i^{[1]} = a_i = \delta \times (1 - \lfloor \frac{i-1}{D/G} \rfloor / G)$ for $i \leqslant \frac{D}{2}$ and $\mu_i^{[1]} = 0$ for $i > \frac{D}{2}$, and $\mu_i^{[2]} = \mu_{k(i)}^{[1]}$, where $k_i = ((i - 1 + \frac{D}{2}) \bmod D) + 1$.
- M10: Similar to $M9$, but with $\Sigma_1 = \mathbf{I}$ and $\Sigma_2 = c\mathbf{I}$.

The second type of model, called the *noise model*, is obtained by adding random noise features to the basic model, where these are randomly permuted with the regular features, as illustrated in Fig. 2. These models are denoted by $M1n$ through $M10n$. Letting D_0 be the number of regular features in the basic model and D_n be the number of noise features, the total number of features is $D = D_0 + D_n$. For both class-conditional distributions, the noise features are modeled as random Gaussian, $N(\vec{0}, \sigma_n^2 \mathbf{I})$.

The models are set-up in such a way that the best feature sets can be predetermined from the distributions, as we next explain.

2.2. Best features

For certain feature-label distributions, we can determine the best feature set, A_{best}, directly from the distribution. We refer to this analytic approach as the *ModelBest* method. It is important to recognize that A_{best} is best in the sense that it gives the lowest (Bayes) error among all feature sets with the same size. In practice, where a different classifier (designed from the data) is used instead

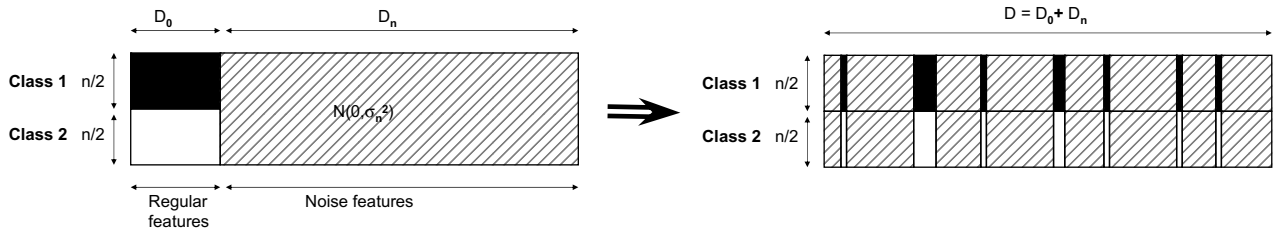

Fig. 2. Illustration of the *noise models*. Noise features are randomly permutated with the regular features, as shown in the figure on the right.

of the Bayes classifier, $\varepsilon_{\text{best}}$, the error of the designed classifier using A_{best}, is not necessarily the lowest.

For analytically determining the best feature set, we focus on the *basic models*, since the additional features for *noise models* are just noise and will not constitute part of A_{best}. The best feature set among all feature sets with size d can be easily derived for models in which all feature are uncorrelated. For models $M1$ and $M2$, since $a_i \geqslant a_j$ whenever $i < j$, A_{best} is comprised of the first d features. This is also true for $M7$ and $M8$, where the Bayes classifier is composed of two hyperplanes normal to $\mu^{[1]}$. In a similar manner, we can find A_{best} for $M9$ and $M10$, as summarized in Table 1.

A_{best} can be also found for models where features are correlated. For a given feature set A of size d, there are two marginal class-conditional distributions for the two classes, each of these being marginal relative to the D-feature class-conditional distribution of the corresponding class. Let μ_1 and μ_2 be the means for these marginal class-conditional distributions, and Σ_{A1} and Σ_{A2} be the corresponding covariance matrices. For models $M3$ and $M5$, $\Sigma_{A1} = \Sigma_{A2} = \Sigma_A$; for $M4$ and $M6$, however, $\Sigma_{A1} = \Sigma_A$ and $\Sigma_{A2} = c \cdot \Sigma_{A1} = c \cdot \Sigma_A$.

For model $M3$, because the priors are equal, the Bayesian boundary is a hyperplane passing through $\frac{1}{2}(\mu_1 + \mu_2)$. Moreover, Σ_A is the same for different A's (with the same size d). We remove the subscript and denote Σ_A as Σ to reflect this fact. The Bayes error is solely determined by the squared Mahalanobis distance,

$$\mathfrak{D} = (\mu_1 - \mu_2)^{\mathrm{T}} \Sigma^{-1} (\mu_1 - \mu_2) \tag{2.2}$$

To see this, let $\Sigma = BB^{\mathrm{T}}$. Since Σ is positive definite, B is invertible. Letting $T = B^{-1}$ be the transformation matrix, the transformed Gaussian has means at $T\mu_1$ and $T\mu_2$, with covariance matrix $T\Sigma T^{\mathrm{T}} = \mathbf{I}$. The squared distance between the two means in the transformed space is

$$\mathfrak{D}' = (T\mu_1 - T\mu_2)^{\mathrm{T}} (T\mu_1 - T\mu_2) \tag{2.3}$$

and a larger Bayes error follows from a smaller $\mathfrak{D}'$, as in model $M1$. Since $\mathfrak{D} = \mathfrak{D}'$, this proves $\mathfrak{D}$ determines the Bayes error in the original space.

If the magnitudes of the a_i's (components in $\vec{\mu}$) are not equal, to find the best d features, we have to compute $\mathfrak{D}$ for all subsets of size d. To circumvent this combinatorial problem, we divide all features into G groups so that in each group all a_i's are equal. This significantly cuts down the number of $\mathfrak{D}$'s that need to be computed since many of them are the same. Table 2 lists the total number of $\mathfrak{D}$'s to be checked for each d for the parameters used in our experiments:

Table 1
Best feature set A_{best} for models $M9$ and $M10$

$\tilde{D} = \frac{D_0}{2}$	
$d = 1$	a_1 or $a_{\tilde{D}+1}$
$d = 2$	$a_1, a_{\tilde{D}+1}$
$d = 3$	$a_1, a_2, a_{\tilde{D}+1}$ or $a_1, a_{\tilde{D}+1}, a_{\tilde{D}+2}$
$d = 4$	$a_1, a_2, a_{\tilde{D}+1}, a_{\tilde{D}+2}$
⋮	⋮
$d = D_0$	$a_1, a_2, \ldots, a_{D_0}$

Table 2
Number of Mahalanobis distances that need to be checked for models $M3$, $M4$, $M5$ and $M6$

$D = 60, G = 6$									
d		d		d		d		d	
1	6	2	21	3	56	4	126	5	252
6	462	7	792	8	1287	9	2002	10	3003
11	4362	12	6152	13	8442	14	11292	15	14748
16	18837	17	23562	18	28897	19	34782	20	41118
21	47762	22	54537	23	61242	24	67662	25	73578
26	78777	27	83062	28	86262	29	88242	30	88913
31	88242	32	86262	33	83062	34	78777	35	73578
36	67662	37	61242	38	54537	39	47762	40	41118
41	34782	42	28897	43	23562	44	18837	45	14748
46	11292	47	8442	48	6152	49	4362	50	3003
51	2002	52	1287	53	792	54	462	55	252
56	126	57	56	58	21	59	6	60	1
Total: 1,771,560									

A similar approach can be taken for $M5$. Now $\mu_1 - \mu_2 = (\delta\delta \cdots \delta)$ remains the same for different feature sets A of size d, but Σ_A can be different. Due to the structure of the covariance matrix in $M5$, we have the same number of values of

$$\mathfrak{D} = (\delta\delta \cdots \delta)^{\mathrm{T}} \Sigma_A^{-1} (\delta\delta \cdots \delta) \tag{2.4}$$

to be checked as in $M3$. The computation is heavier since we will have to evaluate Σ_A^{-1} for each different d-dimensional covariance matrix.

Lastly, the squared distance

$$\mathfrak{D} = (\mu_1 - \mu_2)^{\mathrm{T}} \Sigma_A^{-1} (\mu_1 - \mu_2) \tag{2.5}$$

can also be used to find A_{best} for $M4$ and $M6$. Let $\Sigma_A = BB^{\mathrm{T}}$ and $T = B^{-1}$ be the transformation matrix. Then the transformed Gaussians are $N(T\mu_1, T\Sigma_A T^{\mathrm{T}} = \mathbf{I})$ and $N(T\mu_2, cT\Sigma_A T^{\mathrm{T}} = c\mathbf{I})$. Notice that

$$\begin{aligned}\mathfrak{D}' &= (T\mu_1 - T\mu_2)^{\mathrm{T}} (T\mu_1 - T\mu_2) = (\mu_1 - \mu_2)^{\mathrm{T}} T^{\mathrm{T}} T (\mu_1 - \mu_2) \\ &= (\mu_1 - \mu_2)^{\mathrm{T}} \Sigma_A^{-1} (\mu_1 - \mu_2) = \mathfrak{D}\end{aligned} \tag{2.6}$$

And we know from model $M2$ that $\mathfrak{D}'$ determines the Bayes error. Hence. we can still use the distance, $\mathfrak{D} = (\mu_1 - \mu_2)^{\mathrm{T}} \Sigma_A^{-1} (\mu_1 - \mu_2)$, in the original space to find A_{best}.

2.3. Feature-selection

We implement two wrapper methods. Starting with an empty set A, sequential forward selection (SFS) iteratively adds new features to A, one at a time, so that the new set $A \cup \{f_a\}$ is the best among all $A \cup \{f\}$, $f \notin A$. The problem with SFS is that a feature added to A early may not work well in combination with others but it cannot be removed from A. Sequential forward floating search (SFFS) (Pudil et al., 1994) is introduced to mitigate the problem by "looking-back" for the features already in set A. A feature f_r

is removed from A if $A - \{f_r\}$ is the best (lowest classification error) among all $A - \{f\}, f \in A$, unless f_r, called the "least significant feature", is the most recently added feature. This exclusion continues, one feature at a time, as long as the resulting feature set resulting from removal of the least significant feature is better than the feature set of the same size found earlier in the SFFS procedure.

In addition to the well-known *t-test* method, we have implemented three other filter methods: *Relief*, *CFS* (correlation-based feature-selection), and *MI* (mutual information based feature selection). *Relief* was first introduced by Kira and Rendell (Kira and Rendell, 1992) and further extended by Kononenko and Robnik-Sikonia (Kononenko and Robnik-Sikonia, 1996). The basic idea is that, given a feature-label sample set $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots, (\mathbf{x}_n, y_n)\}$, where y_i is the label for point $\mathbf{x}_i$, for every feature f, each point $\mathbf{x}_i$ is presented and set, $\mathscr{H}_k$, of k-nearest-neighbors (we use $k = 3$ in this study) in the same class and set, $\mathscr{M}_k$, of k-nearest-neighbors in the other class are found. Letting $\mathbf{x}_i^f$ denote the value of feature f at point $\mathbf{x}_i$, the score for feature f is updated using

$$s(f) = s(f) + \sum_{j \in \mathscr{M}_k} g(\mathbf{x}_i^f, \mathbf{x}_j^f) - \sum_{j \in \mathscr{H}_k} g(\mathbf{x}_i^f, \mathbf{x}_j^f) \tag{2.7}$$

where $g(\mathbf{x}_i^f, \mathbf{x}_j^f) = \frac{|\mathbf{x}_i^f - \mathbf{x}_j^f|}{\max(f) - \min(f)}$ and $\max(f)$ and $\min(f)$ are the maximum and minimum values on f, respectively. The features selected are those with the highest scores, $s(f)$.

CFS (Hall, 1999) ranks feature sets by taking into consideration both the predictive ability of the set and redundancy within the set. Specifically, for a feature set A with size d features, it computes the following score:

$$s(A) = \frac{d \cdot \bar{r}_L}{\sqrt{d + d(d-1) \cdot \bar{r}_A}} \tag{2.8}$$

where $\bar{r}_L$ is the average Pearson correlation between features in A and the sample label, and $\bar{r}_A$ is the average correlation between features within A. We use a best-first-search heuristic to avoid the combinatorial problem.

MI is very similar to *CFS*, but instead of computing the Pearson correlation for feature-to-feature and feature-to-label, the mutual information is used to better capture the potential non-linear correlations. Continuous variables are ternarized using k-means before mutual information is computed. A new feature f_a is added to the existing feature set A (with size d) according to the max-relevance-min-redundancy criterion (Peng et al., 2005). To be exact,

$$f_a = \arg\max_{f \notin A} [I(f, L) - \frac{1}{d} \sum_{f_i \in A} I(f_i, f)] \tag{2.9}$$

where L is the sample label.

2.4. Simulation experiment summary

For classification rules, we use linear discrimination analysis (LDA), 3-nearest-neighbor (3NN) and linear support vector machine (LSVM). We summarize our simulation experiments together with the parameters we use in Table 3.

2.5. Patient study

We conduct similar experiments using patient data from a microarray-based classification study that analyzes microarrays prepared with RNA from breast tumor samples from 295 patients (van de Vijver et al., 2002). Using a previously established 70-gene prognosis profile (van't Veer et al., 2002), a prognosis signature based on gene-expression is proposed in (van de Vijver et al., 2002) that correlates well with patient survival data and other clinical measures. Of the 295 microarrays, 115 belong to the "good-prognosis" class and 180 belong to the "poor-prognosis" class.

Table 3
Summary of simulation experiments

Data models	$\mathscr{M}$	$M1 - 10, M1n - M10n$
Classification rules	$\mathscr{R}$	LDA, 3NN, LSVM
Feature-selection methods	$\mathscr{F}$	ModelBest, SFS, SFFS, MI, CFS, T-test, Relief
No. of repetitions	N	500
No. of sample size	n	60
No. of total regular feature	D_0	60
No. of total noise feature dimensions	D_n	0, 30, 60, 120, 540, 1140
No. of total feature dimensions	$D = D_0 + D_n$	60, 90, 120, 180, 600, 1200

$G = 6, \delta = 1, c = 2.25, \rho = 0.4, 0.7$.
$d_{\max} = D_0 - 1, \sigma_n = 0.5$.

The *noise-free* patient model uses intensity gene-expression values associated with the $D = D_0 = 70$ genes. For the *noise* patient model, D_n genes are selected as noise features from the original 24,496 genes not already in the 70-gene pool, and the total number of genes is $D = D_0 + D_n$. For each model, a series of n-point samples S is generated from the 295-point empirical distribution by random selection and the remaining $295 - n$ points are held out for error estimation. Note that the samples are not fully independent on account of overlap resulting from choosing the n sample points from among the same 295 sample points; however, as discussed in (Braga-Neto and Dougherty, 2004), the samples are only weakly dependent, as n is chosen to be rather small in our study. Nonetheless, owing to the dependency, we limit the total number of samples N to 200. It should be noted that since we don't know the distribution of the patient data, the "best" feature set is unknown. Thus we eliminate using "best features" in the patient study. Hence, the procedure is modified as follows:

1. Generate a sample set S with size n and total features D from the 295-point empirical distribution by random selection.
2. Select a size-d feature set A using one of the feature-selection methods $\mathscr{F}$ on S.
3. Design a classifier ψ from S for the feature set A according to the classifier rule $\mathscr{R}$.
4. Compute the error ε_d for ψ using the held out samples.
5. Repeat steps 2 through 4 for $d \in \{1, 2, \ldots, d_{\max}\}$.
6. Repeat steps 1 through 5 N times and compute $\bar{\varepsilon}_d$, the average of the ε_d's.
7. Repeat steps 1 through 6 for different models (*noise-free* model and *noise* with different noise levels), different feature-selection methods $\mathscr{F}$, and different classifier rules $\mathscr{R}$.

The patient study is summarized in Table 4.

3. Results and discussion

We plot $\bar{\varepsilon}_d$-vs-d (average true error versus feature size) for all data models $\mathscr{M}$, classifier rules $\mathscr{R}$, and feature-selection methods

Table 4
Summary of patient experiments

Classification rules	$\mathscr{R}$	LDA, 3NN
Feature-selection methods	$\mathscr{F}$	SFS, SFFS, MI, CFS, T-test, Relief
No. of repetitions	N	200
No. of sample size	n	50
No. of total regular feature	D_0	70
No. of total noise feature dimensions	D_n	0, 35, 70, 140, 630, 1330
No. of total feature dimensions	$D = D_0 + D_n$	70, 105, 140, 210, 700, 1400

$d_{\max} = n - 1$.

$\mathscr{F}$. Here we present some typical results and discuss both simulation and patient studies, while leaving the complete results at the companion website. The vertical bars on x-axis indicate where the $\bar{\varepsilon}_d$ curves peak.

Let us focus on the performances of SFFS and SFS. First, we notice that $\bar{\varepsilon}_d$ for *ModelBest* is not necessarily the lowest and it is often found that SFFS and SFS outperform *ModelBest* when d is large. For example, in model $M1$ using LDA, both SFFS and SFS produce lower errors than *ModelBest* when $d > 20$, as shown in Fig. 3a. As mentioned previously, this is possible because A_{best} found via *ModelBest* is optimal over the underlying distribution and gives the guaranteed lowest error only when the Bayes classifier is used. Even when the classification rule used happens to be in the same class as the Bayes classifier, owing to small sample size, the designed classifier can still be substantially different from the Bayes classifier. In Fig. 3a, the Bayes classifier is linear, the same as LDA. If, however, the number of noise features increases, then it becomes easier for SFFS and SFS to overfit the data and reach a plateau very early, where $\bar{\varepsilon}_d$ is high and does not change much, as in Fig. 3b for LDA, or $\bar{\varepsilon}_d$ has a very small-slope, as in Fig. 3c for 3NN. This behavior is also observed with other data models. In fact, we observe how it evolves as the noise level increases. For instance, in Fig. 3, parts (d) through (i), the performances of SFFS and SFS deteriorate with the addition of more noise features. This behavior occurs for almost all other feature-selection methods as well, but the impact of increased noise is less prominent.

The peaking phenomenon in the presence of feature-selection is extremely varied in its behavior as compared to being given a sequence of features. We do observe that in the range of $[1, d_{\max}]$, there appear three basic curve types for $\bar{\varepsilon}_d$: the convex type as in Fig. 3b (*peaking*), or the small-slope type as in Fig. 3c (*slope*), or the flat curve as in Fig. 3i (*plateau*). It is typical for LDA to have *peaking*, whereas 3NN and LSVM often have either *slope* or *plateau* behavior. Using more features than necessary may be detrimental (as in the *peaking* type), or does not help at all (as in the *plateau* type), or helps only a little (as in the *slope* type). Even in the last case, where a larger d might help some, absent knowledge of the specific distribution, for the potential small gain it may not be prudent to risk peaking. In addition, one may ask whether the limited improvement warrants the additional computation requirement (computation time often increases exponentially with d).

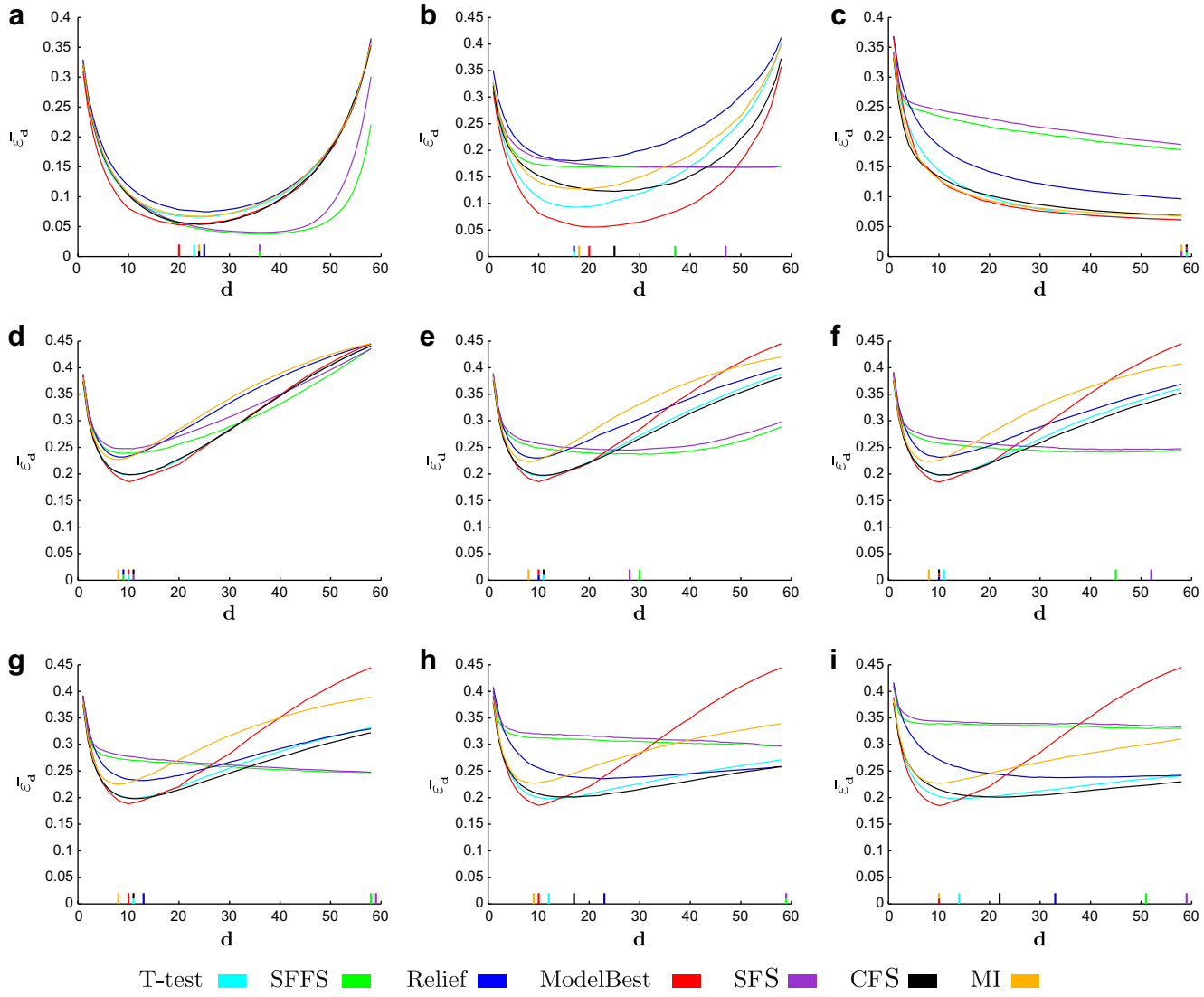

Fig. 3. Performance of SFFS and SFS: (a) LDA classifier, model $M1$, no noise features, and $D_0 = 60$, (b) LDA classifier, model $M1n$, $D_0 = 60$ and $D_n = 1140$ (5% of the total number of features are regular features), (c) 3NN classifier, model $M5n$, $D_0 = 60$ and $D_n = 1140$, (d) 3NN classifier, model $M3$, $D_0 = 60$ and (e)–(i) 3NN classifier, model $M3n$, $D_0 = 60$, and the percentage of noise features out of the total number of features is 33%, 50%, 67%, 90%, and 95%, respectively.

From a practical perspective, especially as pertains to the kind of high-dimensional, small sample settings common in genomics, two salient points can be made. First, it is generally observed that, if peaking occurs for some $d^* \leqslant d_{\max}$, then d^* is smaller for *ModelBest* than for other feature-selection methods, i.e., *ModelBest* peaks early. There are exceptions, but these tend to occur when either different d^*'s are close to each other or when curves are already flattened out over a wide range of d. This means that, in practice, one can generally expect the optimal number of features with feature-selection to be greater than without feature selection. Second, as is clearly the case in Fig 3, parts (g) through (i), it is often the case that the lowest error is achieved with the optimal number of features for *ModelBest*

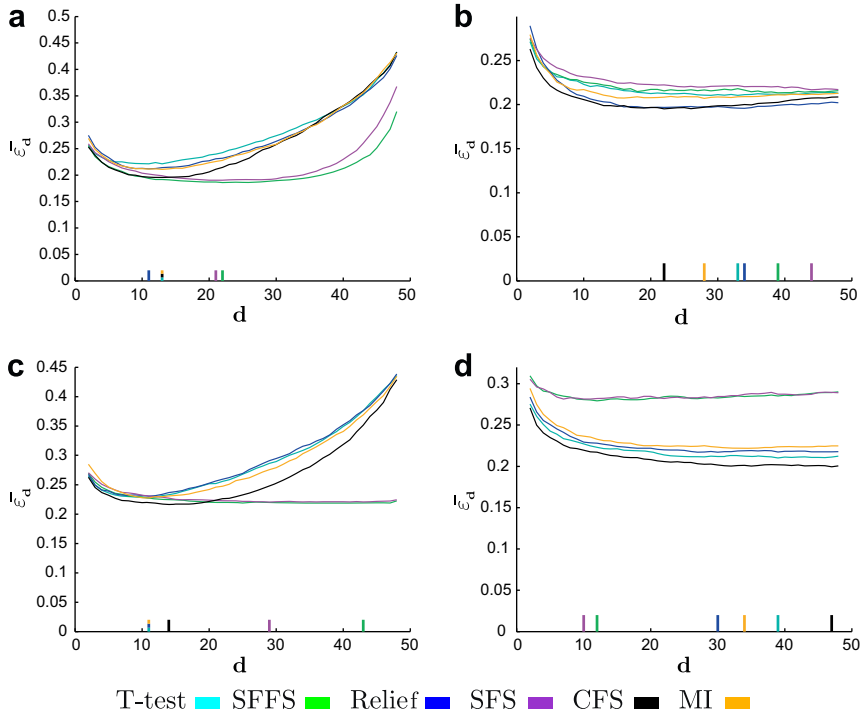

Fig. 4. Patient study for (a) LDA classifier, noise feature number $D_n = 0$, (b) 3NN classifier, $D_n = 0$, (c) LDA classifier, 95% noise features and (d) 3NN classifier, 95% noise features.

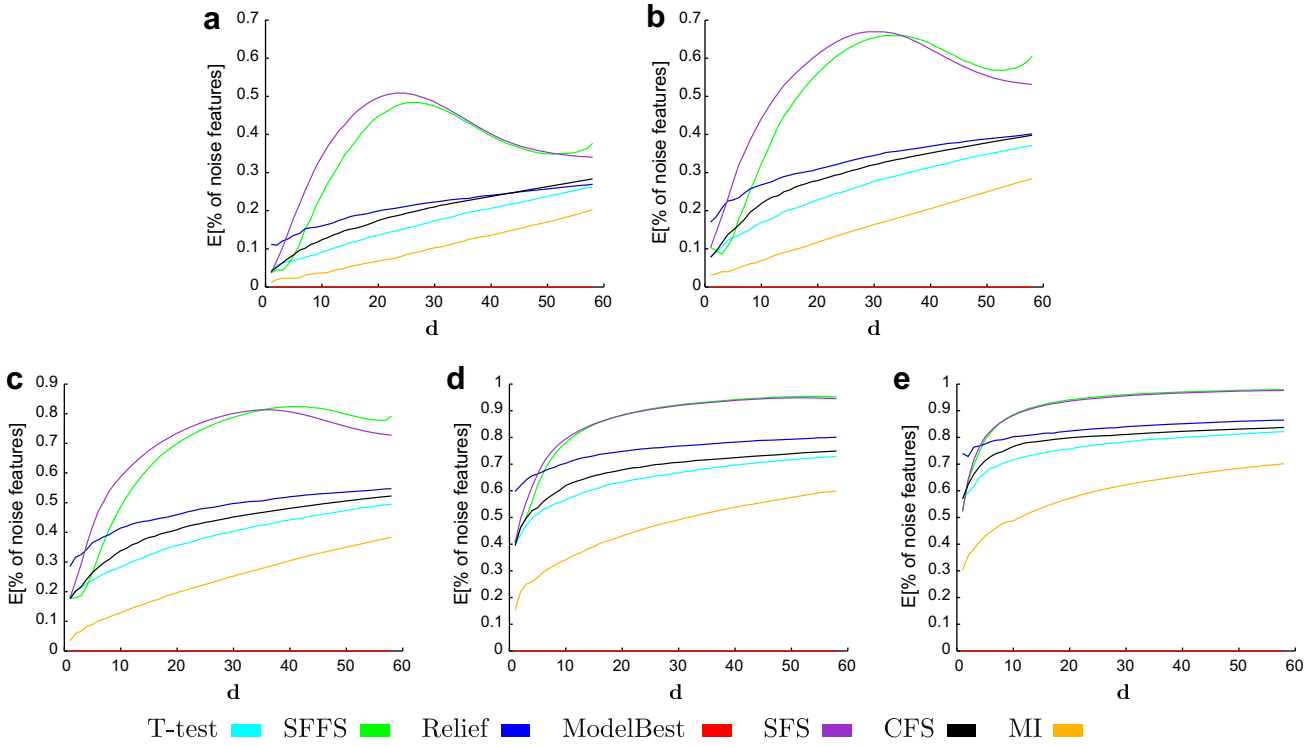

Fig. 5. Expected percentage of noise features as a function of the number of selected features for model $M10n$ with parts (a) through (e) corresponding to $D_n = 30$, 60, 120, 540, and 1140, respectively.

but that the lowest error of *ModelBest* is not robust relative to using too many features, whereas the lowest errors of the various feature-selection methods, while higher than the lowest error for *ModelBest*, are much more robust relative to using too many features. This has significant practical import because the detrimental effect of using too many features is not as great as one might conclude from looking at classical papers that pre-set the order of the features. Owing to the varied results across feature-label models, classifier models, and feature-selection algorithms, clearly one must be prudent in making this statement; nonetheless, to the extent that the loss of performance from using too many features is mitigated in a number of cases, this might mean that some of the large feature sets reported in the genomics literature are not so bad as one might first assume. Perhaps it would be wise for those interested in particular published results to go back and investigate the peaking phenomenon in those settings.

Similar results are found for the patient study, four examples of which are shown in Fig. 4. Again, $\bar{\varepsilon}_d$ either remains almost constant as d gets larger or shoots up significantly, e.g., *filter* methods for LDA in Fig. 4a and c. A feature set with size $d \approx 10$ seems to suffice for classification purposes for this data set.

For models including noise features, an obvious issue in feature-selection performance, and therefore the degree of peaking, is the percentage of noise features selected. To examine this issue we have considered LDA classification for models $M1n$ through $M10n$, using different proportions of noise features, $D_0 = 60$ and $D_n = 30, 60, 120, 540$, and 1140. The graphs of the expected percentage of noise features as a function of the number of selected features are given for all ten models on the companion website. Here, Fig. 5 shows the graphs for model $M10n$ with parts (a) through (e) corresponding to $D_n = 30, 60, 120, 540$, and 1140, respectively. For filter-type feature-selection algorithms, the expected percentage of noise features increases monotonically with the number, d, of features. For very small numbers of features, SFS and SFFS perform comparatively well with respect to the expected percentage of noise features; however, they deteriorate rapidly with increasing numbers of features. When the number of noise features is no larger than 120, they eventually begin to improve for larger d relative to the expected percentage of noise features. The most important point to glean from Fig. 5 is that, as the proportion of noise features increases from part (a) through part (e), the expected percentage of noise features increases across all numbers of features and feature-selection algorithms. The number of informative features is simply being overwhelmed by the number of noise features.

4. Conclusion

We have studied the peaking phenomenon in the presence of feature selection for various simulation models and a set of breast-cancer data. The $\bar{\varepsilon}_d$-vs-d curves for different feature selection methods are found to be falling into one of three categories: peaking, slope or plateau. Thus, if feature-selection algorithms are used with classifier design, which is virtually always true in practice in high-dimensional situations, one should avoid using too many features because, at best, doing so hardly helps and in many cases harms classification accuracy. This is especially true for those models in which the number of noise features is relatively high, a situation likely to be encountered in high-dimensional settings, where the useful features are mixed with many more noise features. This observation agrees well with what we find in the breast-cancer patient study. It is important to recognize that a perusal of the full set of experiments done in this study shows the highly complex behavior of error curves in the presence of feature-selection and one should be wary of applying peaking results found in the absence of feature-selection to settings in which feature-selection is being employed; indeed, as we have pointed out, in many cases feature-selection mitigates the steepness of the peaking phenomenon, albeit, at the cost of poorer optimal performance.

Acknowledgement

This work has been sponsored in part by the National Science Foundation (CCF-0634794) and the National Cancer Institute (R01 CA-104620). The authors would also like to thank the High-Performance Biocomputing Center of TGen for providing the clustered computing resources used in this study.

References

Ambroise, C., McLachlan, G.J., 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. Proc. Natl. Acad. Sci. USA 99 (10), 6562–6566.
Braga-Neto, U.M., Dougherty, E.R., 2004. Is cross-validation valid for small sample microarray classification? Bioinformatics 20 (3), 374–380.
Cover, T.M., Van Campenhout, J., 1977. On the possible orderings in the measurement selection problem. IEEE Trans. Systems Man Cybernet. 7 (9), 657–661.
Hall, M.A., 1999. Correlation-based feature-selection for machine learning. PhD Thesis, University of Waikato.
Hua, J., Xiong, Z., Dougherty, E.R., 2005a. Determination of the optimal number of features for quadratic discriminant analysis via the normal approximation to the discriminant distribution. Pattern Recognition 38 (3), 403–421.
Hua, J., Xiong, Z., Lowey, J., Suh, E., Dougherty, E.R., 2005b. Optimal number of features as a function of sample size for various classification rules. Bioinformatics 21 (8), 1509–1515.
Hughes, G.F., 1968. On mean accuracy of statistical pattern recognizers. IEEE Trans. Inform. Theory It14 (1), 55–63.
Jain, A.K., Waller, W.G., 1978. On the optimal number of features in the classification of multivariate Gaussian data. Pattern Recognition 10 (5–6), 365–374.
Kira, K., Rendell, L.A., 1992. A practical approach to feature-selection. In: Sleeman, D., Edwards, P. (Eds.), Proc. Ninth Internat. Workshop on Machine Learning. Morgan Kaufmann Publishers Inc., Aberdeen, Scotland, United Kingdom, pp. 249–256.
Kononenko, I., Robnik-Sikonia, M., 1996. Relief for estimation and discretization of attributes in classification. In: Ramsay, A. (Ed.), Artificial Intelligence: Methodology, Systems, Applications. IOS Press, pp. 31–40.
Li, L., Zhang, J., Neal, R.M., 2007. A method for avoiding bias from feature-selection with application to naive bayes classification models. Technical Report 0705, Department of Statistics, University of Toronto.
Murray, G.D., 1977. A cautionary note on selection of variables in discriminant analysis. Appl. Statist. 26 (3), 246–250.
Navot, A., Gilad-Bachrach, R., Navot, Y., Tishby, N., 2006. Is feature-selection still necessary? In: Saunders, C., Grobelnik, M., Gunn, S.R., Shawe-Taylor, J. (Eds.), Subspace, Latent Structure and Feature-Selection, Statistical and Optimization, Perspectives Workshop, SLSFS 2005, Bohinj, Slovenia, February 23–25, 2005, Revised Selected Papers. Lecture Notes in Computer Science, vol. 3940. Springer, Berlin, New York, pp. 127–138.
Peng, H., Long, F., Ding, C., 2005. Feature-selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans. Pattern Anal. Machine Intell. 27 (8), 1226–1238.
Pudil, P., Novovicova, J., Kittler, J., 1994. Floating search methods in feature-selection. Pattern Recognition Lett. 15 (11), 1119–1125.
Raudys, S.J., 1979. Determination of optimal dimensionality in statistical pattern classification. Pattern Recognition 11 (4), 263–270.
Raudys, S.J., Jain, A.K., 1991. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Trans. Pattern Anal. Machine Intell. 13 (3), 252–264.
Sima, C., Attoor, S., Brag-Neto, U., Lowey, J., Suh, E., Dougherty, E.R., 2005. Impact of error estimation on feature-selection. Pattern Recognition 38 (12), 2472–2482.
Singhi, S.K., Liu, H., 2006. Feature subset selection bias for classification learning. In: The 23rd Internat. Conf. on Machine Learning. ACM Internat. Conf. Proceeding Series, vol. 148. ACM, Pittsburgh, Pennsylvania, pp. 849–856.
Trunk, G.V., 1979. Problem of dimensionality – simple example. IEEE Trans. Pattern Anal. Machine Intell. 1 (3), 306–307.
van de Vijver, M.J., He, Y.D., van't Veer, L.J., Dai, H., Hart, A.A., Voskuil, D.W., Schreiber, G.J., Peterse, J.L., Roberts, C., Marton, M.J., Parrish, M., Atsma, D., Witteveen, A., Glas, A., Delahaye, L., van der Velde, T., Bartelink, H., Rodenhuis, S., Rutgers, E.T., Friend, S.H., Bernards, R., 2002. A gene-expression signature as a predictor of survival in breast-cancer. N. Engl. J. Med. 347 (25), 1999–2009.
van't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R., Friend, S.H., 2002. Gene-expression profiling predicts clinical outcome of breast-cancer. Nature 415 (6871), 530–536.
Zhou, X., Mao, K.Z., 2006. The ties problem resulting from counting-based error estimators and its impact on gene selection algorithms. Bioinformatics 22 (20), 2507–2515.