Effect of finite sample size on feature selection and classification: A simulation study

Ted W. Way, Berkman Sahiner,[a)] Lubomir M. Hadjiiski, and Heang-Ping Chan
Department of Radiology, University of Michigan, Ann Arbor, Michigan 48109-5842

(Received 26 June 2009; revised 27 October 2009; accepted for publication 30 November 2009; published 28 January 2010)

Purpose: The small number of samples available for training and testing is often the limiting factor in finding the most effective features and designing an optimal computer-aided diagnosis (CAD) system. Training on a limited set of samples introduces bias and variance in the performance of a CAD system relative to that trained with an infinite sample size. In this work, the authors conducted a simulation study to evaluate the performances of various combinations of classifiers and feature selection techniques and their dependence on the class distribution, dimensionality, and the training sample size. The understanding of these relationships will facilitate development of effective CAD systems under the constraint of limited available samples.

Methods: Three feature selection techniques, the stepwise feature selection (SFS), sequential floating forward search (SFFS), and principal component analysis (PCA), and two commonly used classifiers, Fisher's linear discriminant analysis (LDA) and support vector machine (SVM), were investigated. Samples were drawn from multidimensional feature spaces of multivariate Gaussian distributions with equal or unequal covariance matrices and unequal means, and with equal covariance matrices and unequal means estimated from a clinical data set. Classifier performance was quantified by the area under the receiver operating characteristic curve A_z. The mean A_z values obtained by resubstitution and hold-out methods were evaluated for training sample sizes ranging from 15 to 100 per class. The number of simulated features available for selection was chosen to be 50, 100, and 200.

Results: It was found that the relative performance of the different combinations of classifier and feature selection method depends on the feature space distributions, the dimensionality, and the available training sample sizes. The LDA and SVM with radial kernel performed similarly for most of the conditions evaluated in this study, although the SVM classifier showed a slightly higher hold-out performance than LDA for some conditions and vice versa for other conditions. PCA was comparable to or better than SFS and SFFS for LDA at small samples sizes, but inferior for SVM with polynomial kernel. For the class distributions simulated from clinical data, PCA did not show advantages over the other two feature selection methods. Under this condition, the SVM with radial kernel performed better than the LDA when few training samples were available, while LDA performed better when a large number of training samples were available.

Conclusions: None of the investigated feature selection-classifier combinations provided consistently superior performance under the studied conditions for different sample sizes and feature space distributions. In general, the SFFS method was comparable to the SFS method while PCA may have an advantage for Gaussian feature spaces with unequal covariance matrices. The performance of the SVM with radial kernel was better than, or comparable to, that of the SVM with polynomial kernel under most conditions studied. [DOI: 10.1118/1.3284974]

Key words: feature selection, linear discriminant analysis, support vector machines, sample size effect

I. INTRODUCTION

Advances in computer processing power, memory capacity, imaging technologies, and image processing algorithms have greatly improved the diagnostic information available to radiologists. Image processing and analysis tools allow computers to aid radiologists in previously time-consuming tasks. Computer-aided detection and diagnosis (CAD) software further facilitates image interpretation. Computer-aided detection systems mark suspicious areas on images that radiologists may have overlooked to prompt them to examine that area more carefully. Computer-aided diagnosis software for cancer provides a malignancy estimate of suspicious tissue.

For these computer aids to be effective, however, a CAD system would need to extract salient features from the images, choose only the features that can discriminate between classes, and accurately classify previously unseen samples. Ideally, there would be a large number of training samples to design the CAD system, but it is expensive and time-consuming to collect case samples with ground truth. The

development of CAD systems for automatic detection and diagnosis of lung nodules on computed tomography (CT) can serve as an example. For detection studies, the gold standard for determining whether abnormal-appearing tissue is considered a nodule is often determined by a consensus among radiologists, yet considerable interobserver variability makes the truth uncertain. For diagnosis studies, a nodule is considered benign when it shows no change for at least 2 yr. Determination of malignancy often requires biopsy. If multiple nodules are present in the lungs, biopsy may not be performed for every nodule because of the risks and expenses. These factors limit the number of samples available to train, validate, and test CAD systems.

An important issue in CAD system development is whether the performance on training data is generalizable to the population at large. It is therefore useful to estimate the bias and variance of the classifier performance on previously unseen samples. This would allow users to predict the performance when the CAD system is applied to unknown cases in clinical practice. Classifiers for the differentiation of true and false lesions or for the differentiation of malignant and benign lesions are some of the main components in a CAD system. Studies of sample size effects on classifier design exemplify similar problems in development of CAD systems. Previous simulation studies have focused on the effect of finite sample size on classifier performance when the samples were drawn from multivariate Gaussian distributions of various dimensionalities. Resubstitution and hold-out methods were used for estimating classifier performance. In the resubstitution method, the classifier performance is measured by applying the classifier to the training samples that have been used to design it. In the hold-out method, the samples are partitioned into training and test samples. The classifier is designed with only the training samples and then evaluated on the independent test samples. Chan *et al.*[1] compared the sample size effects on the design of the linear discriminant, the quadratic discriminant, and the backpropagation artificial neural networks (ANNs). For feature spaces of Gaussian distributions with unequal covariance matrices and three to 15 dimensions, the linear discriminant analysis (LDA) classifier was inferior to the quadratic discriminant or the ANN when there were a large number of training samples. However, with a small number of training samples available, a simpler classifier such as the LDA or ANN with few nodes may be preferred. A small sample size becomes even more limiting when one has to select the most effective features from a large pool of available features using the same small sample set. Sahiner *et al.*[2] investigated the effect of sample size, number of available features, and the parameters for stepwise feature selection (SFS) on LDA performance. They found that the resubstitution estimate was always optimistically biased, except when there were too few features. The hold-out estimate was always pessimistically biased when the classifier was trained on only the training samples.

In recent studies, Sahiner *et al.* investigated the bias and variance of various resampling methods in predicting the performance of a classifier for unknown samples when the classifier is trained with a finite sample size. Two classifiers, Fisher's LDA (Ref. 3) and backpropagation ANN (Ref. 4) were evaluated. Under their study conditions, they found that the prediction accuracy depends strongly on the resampling method, especially for large feature dimensionality and small sample sizes. Li and Doi[5] performed a simulation study and proposed an automated threshold selection method to minimize the overtraining effect in rule-based classifier design. Li and Doi[5] also performed another study[6] to compare evaluation methods for CAD systems such as the bias of the estimated performance, the generalization performance, and the uniqueness of the CAD scheme. Beiden *et al.*[7] focused on the variance of competing classifiers. They concluded that in comparing various classifiers, the variance contributed by the finite training sample is the dominant component. This is opposed to the conventional wisdom that the finite training sample size contributed to the bias on the performance measures, while the variance is mainly determined by the finite number of test samples.

The interaction between the feature selection method and the classifier used will also influence the training of a classifier. Some combinations of methods may perform better than others given a small sample size, some may generalize better to unknown samples, and others may result in lower variance. Jain and Zongker[8] compared various feature selection algorithms and concluded that the sequential forward floating search[9] (SFFS) method performed better than other methods. Kudo and Sklansky[10] also concluded that SFFS was effective for small-scale and medium-scale problems while genetic algorithms would be better suited for large-scale problems.

The goal of this study is to investigate combinations of feature selection techniques and classifiers and to compare their performance on two classes of data drawn from multivariate Gaussian distributions with unequal means and either equal or unequal covariance matrices. The effects of the covariance matrices, finite sample size, and the dimensionality of the feature spaces on the performance of the classifier were studied. Although the performance of feature selection methods and classifiers have been investigated extensively in the literature, there are only limited studies on combinations of these two important processes for various feature selection techniques and classifiers under the condition of limited training sample size. Sima and Dougherty[11] performed a simulation study in which LDA, support vector machine (SVM), and three nearest neighbor methods were used as classifiers, and SFFS and the *t*-test were used for feature selection. Since their focus was the comparison between the performance of the optimal and selected feature sets, they did not present a comparison of different combinations of the classifiers and feature selection methods. In addition, the effect of the training sample size was not studied. Hua *et al.*[12] used LDA, linear SVM, and nearest neighbor classifiers with ten feature selection methods in a simulation study and with real data. Most of the presented results were based on the LDA, and the comparison of different combinations was not presented in detail. Lee *et al.*[13] compared a number of classification methods with three gene selection methods on

seven gene expression data sets. All three selection methods in their study were univariate, i.e., were based on measures of the performance of individual genes. The performance of combined features was not considered at the gene selection stage. The current study investigated the relationship between feature selection methods and classifiers in multivariate normal feature spaces over a range of training sample size and dimensionality of the feature space. Although the conditions that can be covered in a single study are still limited, this relatively systematic study of representative feature selection methods and classifiers provides further understanding of the issues for classifier design. The information may serve as a guide in future CAD system development and prompt further investigations in these important areas.

II. METHODS AND MATERIALS

To train a classifier in a CAD system, the first step is often feature extraction from the case samples. Since it is not known *a priori* whether the computer-extracted features may be useful for the classification task at hand, a large number of possible features are often extracted and a feature selection method is used to choose the most effective features. A classifier is then built using the selected features as input predictor variables. Both the feature selection and the classifier parameters should be trained on the training samples only. The performance of the trained classifier on unknown cases is then estimated on the independent samples that have been held out for testing.

II.A. Class distributions

In this simulation study, the training and test samples for the two classes were drawn randomly from two multivariate Gaussian distributions of three different types: (1) Equal covariance matrices with unequal means, (2) unequal covariance matrices with unequal means, and (3) equal covariance matrices estimated from clinical data with unequal means. While a number of previous studies that investigated feature selection performance also used simulated Gaussian data,[2,8,9,14] a few others simulated a mixture of Gaussians[15,16] and Boolean feature spaces.[17,18] Although clinical data may not follow any of these idealized distributions, we chose to use Gaussian distributions because they are commonly used in both simulation studies and theoretical analyses of classifier performance in pattern recognition literature.

A set of N_s samples was generated from each class distribution using a random number generator. The details of the two classes are described below. This set was then randomly partitioned into N_{train} training and N_{test} test samples per class. We varied N_{train} and fixed N_{test} to be 100 per class for a given feature space to study the effect of training sample size on classifier performance. For a given number of training and testing samples, 1000 experiments were performed with a new set of samples generated for each experiment. Keeping N_{test} fixed for different experiments allowed us to directly investigate the dependence of the variance of the performance measure on the number of training samples, without the confounding effects of the variation in the number of test samples. The resubstitution and hold-out test performances of the classifier were quantified by the area under the receiver operating characteristic curve A_z. The mean and the variance of the resubstitution and hold-out test A_z for the given sample size were estimated from the 1000 experiments.

II.A.1. Equal covariance matrices and unequal means

The first condition simulated two classes with multivariate Gaussian distributions and equal covariance matrices. Without loss of generality, we used two identity matrices because a common arbitrary covariance matrix for both classes can be simultaneously diagonalized and the variances of the individual feature components normalized to unity.[19] The mean feature vector of the first class was zero, $\mu_1=0$, and the difference in the class means, $\Delta\mu(i)$, between the two classes for feature i was given by[2]

$$\Delta\mu(i) = \mu_2(i) - \mu_1(i) = \alpha\beta^i, \quad i = 1, \ldots, M \quad \text{and} \quad \beta < 1, \tag{1}$$

where M is the dimensionality of the available feature space from which a number of features may be selected. The squared Mahalanobis distance between the two classes Δ was computed as[20]

$$\Delta = \frac{\alpha^2\beta^2}{1-\beta^2}(1-\beta^{2M}) \tag{2}$$

since all the diagonal values of the covariance matrix were 1. The parameter β was set to be 0.9 and α was chosen such that $\Delta=3.0$. Feature i therefore has decreasing ability to separate the two classes as i increases. The specific form of the features and the values of these parameters were not critical for the purpose of this simulation study; they were designed to generate a set of features that have varying discriminatory powers to distinguish the two classes. For the equal covariance matrix condition, the Mahalanobis distance can be used to determine the ideal A_z value of the optimal classifier trained and tested with the true (infinite-sized) population,[1] denoted as $A_z(\infty)$. In this study, the Mahalanobis distance was selected such that $A_z(\infty)=0.89$, which is representative of the range of A_z values achieved in CAD literature. The classification accuracy for $M=50$, 100, and 200 was investigated.

II.A.2. Unequal covariance matrices and unequal means

This condition simulated two classes that follow underlying Gaussian distributions with different covariance matrices. The covariance matrix of the first class was diagonalized and scaled as the identity matrix $\Sigma_1=I$, with $\mu_1=0$. The covariance matrix of the second class Σ_2 was simultaneously diagonalized such that it had eigenvalues $v_i, i=1, \ldots, M$, where M is the dimensionality of the feature space available for selection. The values of v_i were generated by

$$v_i = 1 + \varepsilon(\gamma^{M-i} - 1), \quad i = 1, \ldots, M, \quad \varepsilon = \frac{v_{\max} - 1}{-1 + \gamma^{M-1}}, \qquad (3)$$

where $\gamma = 1.5$, $v_{\max} = v_1 = 3$, and the smallest eigenvalue $v_{\min} = v_M$ was set to 1. The eigenvalues of the covariance matrix for the second class therefore decreased exponentially from $v_{\max}$ to 1 as the feature number changed from 1 to M. The values of the mean vector of the second class μ_2 were calculated according to Eq. (1), where $\beta = 0.9$. For the unequal covariance matrix condition, there is no closed-form solution that relates the mean and covariance matrices of the class distributions to $A_z(\infty)$. However, a close approximation for $A_z(\infty)$ in terms of the Bhattacharyya distance[19,21] has been derived.[22] In this study, the value of α in Eq. (1) was chosen such that the Bhattacharyya distance between the two classes was 3/8, which corresponded to $A_z(\infty) \approx 0.89$. With the selected values of α and β, the squared Mahalanobis distance was 1.66, which was lower than that in the equal covariance matrix condition. The nonidentity covariance matrix was designed such that the greatest separation in the mean value corresponded with the greatest eigenvalue in the covariance matrix. Since our goal was to compare the performance of various features selection methods and classifiers, the specific values of $v_{\max}$ and $v_{\min}$ were not critical.

II.A.3. Equal covariance matrices based on clinical data and unequal means

To simulate features from clinical data that may be encountered by a CAD system, we first extracted features from volumes-of-interest containing lung nodules from CT scans. These features were extracted with the goal of classifying the lung nodules as malignant or benign.[23] They included morphological features such as volume and perimeter, in addition to gray-level statistics, texture features from run-length statistics,[24,25] gradient field, and radii features. The means and covariance matrices of each class were estimated from a database of 124 malignant and 132 benign nodules. These estimated means and covariance matrices were assumed to be the true underlying multivariate Gaussian distributions of the population for this study. We assumed that the two classes had the same multivariate Gaussian distribution with covariance matrix $\Sigma = (\Sigma_1 + \Sigma_2)/2$, where Σ_1 and Σ_2 were estimated from the malignant and benign classes, respectively, of the clinical data.

II.B. Feature selection methods

Typical feature selection strategies include the "top-down" and "bottom-up" methods. Marill and Green[26] introduced the top-down method, which is initialized with the entire feature space. Features are removed after certain criteria have been met to obtain the set of remaining features to be used. Its counterpart is the bottom-up method, which is initialized with the empty set, and features are added until certain criteria have been met.[27] The disadvantage of these methods is the "nesting effect," in which features removed are no longer considered or features added cannot be removed. The SFS and SFFS methods were designed to overcome the nesting effect.

II.B.1. SFS

We used a linear model and employed the Wilks' lambda[21] based on the outcome of the linear model as the feature selection criterion in SFS. Initially, all features are tested to find the one that provides the best value of the selection criterion. At each subsequent step, every feature that has not been selected is evaluated to determine how much the feature can improve the selection criterion when it is combined with the set of already selected features. Wilks' lambda is defined as the ratio of the within-group sum of squares to the total sum of squares

$$\lambda_d = \frac{\Sigma_{i \in \text{class1}} (h^{(d)}(X_i) - m_1^{(d)})^2 + \Sigma_{i \in \text{class2}} (h^{(d)}(X_i) - m_2^{(d)})^2}{\Sigma_{i=1}^{N} (h^{(d)}(X_i) - m^{(d)})^2}, \qquad (4)$$

where d is the dimensionality of the selected feature subspace, $h^{(d)}(X_i)$ is the discriminant score for the input vector X_i consisting of the selected features for case i, $h^{(d)}(X_i) = b^T X_i + b_0$, with $b^T = [b_1, b_2, \ldots, b_d]$ and b_0 being the LDA coefficients, $m_1^{(d)}$ and $m_2^{(d)}$ are the means of the discriminant scores for classes 1 and 2, respectively, $m^{(d)}$ is the mean of the discriminant scores for both classes, and N is the number of available training samples. The smaller the value of Wilks' lambda, the smaller the spread within each class relative to the spread of the entire sample, indicating that the separation of the two classes is larger and better classification can be achieved.

The significance of the change in Wilks' lambda for a new feature entered into the analysis is based on F statistics.[28,29] To determine whether a feature should be included when d features have already been selected, the *F -to-enter* value is calculated[30] for each feature that has not been selected

$$F = (N - d - 2)\left(\frac{\lambda_d}{\lambda_{d+1}} - 1\right), \qquad (5)$$

where λ_d and λ_{d+1} are the Wilks' lambda values before and after entering the feature to the pool of selected features. The feature with the largest F-to-enter value is added to the selected features if its value is higher than a threshold F_{in}. A lower F_{in} threshold means that it is easier to add more features, resulting in a larger set of selected features. After a feature is entered, each feature in the selected pool is tested for removal by calculating the *F -to-remove* value, which is defined similarly to F-to-enter. The feature with the smallest F-to-remove value that is also lower than a threshold F_{out} is removed. A lower F_{out} makes it more difficult to remove features, which will lead to a larger set of selected features. This process of entering and removing a feature is repeated until no more features satisfy the criteria for entry or removal. Another threshold is the tolerance term, which prevents a feature from being entered when it is highly corre-

lated with the already selected features, even if the feature satisfies the F_{in} threshold. Because the thresholds are not known *a priori*, and it is not practical to search through all combinations, we set $F_{out}=F_{in}-1$, where F_{in} was varied from 2 to 7 to cover a reasonable range of values, and the tolerance threshold was fixed at 0.001. These thresholds result in a wide range of the number of features selected, allowing us to demonstrate the effect of finite sample size on feature selection and classifier performance.

II.B.2. SFFS

A disadvantage of SFS is that it only allows one feature to be added or discarded at a time. The plus-*l*-minus-*r* method[31] allows the addition of *l* or removal of *r* features at a time, but there is no theoretical way to predict the best *l* and *r* values. Pudil *et al.*[9] introduced the floating search method, which is a suboptimal search method that assesses the performance of combinations of features. The number of features added or removed at each step changes dynamically, and a predefined number of desired features control the stopping criterion.

The SFFS method is initialized with the best performing combination of two features based on the Mahanalobis distance between the two classes. A table stores the best performing feature combinations of cardinalities of 1 through a number beyond the total number of desired features plus delta. As features are added and removed, the performance of features is assessed. If a better performing combination of the same cardinality is found, then that combination is updated in the table. The procedure terminates when the number of selected features reaches the predetermined number of desired features plus delta. This allows the SFFS algorithm to search for combinations of features of cardinality beyond the desired number of features. The best feature combination corresponding to the desired cardinality can then be chosen. Interested readers are referred to Pudil *et al.*[9] for the detailed procedure of the SFFS method. We chose to examine the performance of 5, 8, 11, 14, 17, and 20 desired features and a delta value of 5 in this study because this range encompassed the number of features selected by the SFS method for all but a few extreme cases under our simulation conditions.

II.B.3. PCA

PCA transforms a number of correlated variables into a number of uncorrelated variables, i.e., the principal components. It performs eigenvalue decomposition of the estimated covariance matrix of the features, projecting the multivariate feature vectors onto the space spanned by the eigenvectors. The order of a principal component represents its importance in accounting for the variance in the data set. The dimensionality of the feature space is reduced by retaining the lower-order principal components that are most important while ignoring the higher-order ones. Retaining only the lower-order principal components is essentially equivalent to approximating the data by a linear subspace using the mean squared error criterion.[32] The orders of 5, 8, 11, 14, 17, and 20 were selected for this study, which spanned a similar range as the number of features selected by SFS, except when $F_{in}=3$ or 2.

II.C. Classification methods

A large number of linear and nonlinear classifiers have been developed in the literature for various pattern recognition and machine learning problems. We selected two commonly used classifiers, Fisher's LDA and the SVM with two different kernels, as examples of linear and nonlinear classifiers to compare their performance in combination with the SFS, SFFS, and PCA methods.

II.C.1. LDA

The LDA classifier uses the means and covariance matrices of the two class distributions to calculate a linear decision boundary separating the two classes. The classifier is described as[19,33]

$$h_l(X) = (\mu_2 - \mu_1)^T \bar{\Sigma}^{-1} X + \frac{1}{2}(\mu_1^T \bar{\Sigma}^{-1} \mu_1 - \mu_2^T \bar{\Sigma}^{-1} \mu_2), \qquad (6)$$

where $\bar{\Sigma}=(\Sigma_1+\Sigma_2)/2$ and X is the feature vector. The means and covariance matrices have to be estimated from the available training samples. A nonlinear transformation of the sample means and covariance matrices results in the LDA coefficients. The LDA coefficients are then linearly combined with the test data to obtain the discriminant scores, which are transformed nonlinearly into a performance measure. The variances due to the estimated parameters propagate to the mean classifier performance, resulting in a bias through the second derivative of the transformation function.

It is known that the LDA classifier is optimal for multivariate normal distributions with equal covariance matrices. The classifier performance in the limit of large training samples can be calculated by the Mahalanobis distance

$$A_z = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\sqrt{\Delta/2}} e^{-u^2/2} du. \qquad (7)$$

In this study, we set $\Delta=3$ for the equal covariance matrix condition, and thus the maximum achievable $A_z(\infty)$ by the optimal linear discriminant is 0.89 in the limit of a large number of training samples. For the unequal covariance matrix condition, we set $\Delta=1.66$ for the chosen feature space, which corresponds to $A_z=0.82$ using Eq. (7) and $A_z(\infty) \approx 0.89$ in terms of the Bhattacharyya distance[19,21] for this second simulation condition. For the third condition that used the clinical data with the estimated means and assumed equal covariance matrices, it was calculated that $\Delta=4.91$, which corresponded to $A_z=0.94$.

II.C.2. SVM

The SVM works similarly to the LDA by constructing a decision hyperplane to separate classes using training data. A brief overview of the SVM is given here, with more details in the literature.[34] Geometrically, the SVM maps the original data to a higher dimension space H via a kernel K. A deci-

TABLE I. Summary of the combinations of feature selection method and classifier that resulted in the lowest bias on the hold-out performance under the experimental conditions in this study.

Covariance matrices	Mean	Large training sample size	Small training sample size
Equal	**Unequal**	SFS or SFFS and LDA or SVM(rad)	PCA and LDA
Unequal	**Unequal**	PCA and LDA or SVM(rad)	PCA and LDA
Clinical	**Unequal**	SFS or SFFS and LDA	SFS or SFFS and SVM(rad)

sion hyperplane is constructed in this higher dimension such that the distance between the training samples of both classes and the hyperplane is maximized. This distance between a training sample and the hyperplane is called the margin, and the SVM calculates the hyperplane with the largest margin.

Suppose we have labeled training samples $\{x_i, y_i\}$, $i=1\ldots N$, $y_i \in \{-1,1\}$, $x_i \in R^d$, where N is the number of samples and d is the dimensionality of the selected feature space (number of selected features). In the SVM formulation, the data appear in the form of dot products $x_i \cdot x_j$. First, the SVM algorithm uses a mapping Φ to transform the data to some other Euclidean space H $\Phi: R^d \mapsto H$. The transformation depends only on the dot products in H of the form $\Phi(x_i)\cdot\Phi(x_j)$. There exist kernel functions K such that $K(x_i,x_j)=\Phi(x_i)\cdot\Phi(x_j)$, and only K is needed in the training algorithm. No explicit knowledge of Φ is necessary. Various kernels have been investigated in the literature, and we chose two commonly used ones, the radial and polynomial kernels.[35–37] In the following, the SVM with the radial and polynomial kernels are referred to as SVM(rad) and SVM(poly), respectively. For the radial kernel, we set $\gamma=0.01$ and capacity C to 1, while for the polynomial kernel, we set degree 2 and $C=1$. These values were chosen experimentally based on classification studies on the clinical lung nodule data set discussed in Sec. II A 3. We implemented the SVM with the freely available MYSVM (Ref. 38) software.

II.D. Simulation study

The number of training samples per class randomly drawn from the class distributions was 15, 20, 30, 40, 50, 60, 80, 90, and 95. The number of test samples per class was fixed at 100 so that the variances in the hold-out classification performance due to the test set size are kept relatively constant. The dimensionality of the input feature spaces M were chosen to be 50, 100, and 200.

Combinations of the three feature selection methods (SFS, SFFS, and PCA) and three classifiers [LDA, SVM(rad), and SVM(poly)] were trained and tested on the available samples. For each combination, there were three different types of feature space distributions, as discussed above. The resulting resubstitution and hold-out A_z values, in addition to the variances were compared.

III. RESULTS

The results of the simulation study for various combinations of feature selection and classification methods are described below. For a given number of training samples, the mean A_z obtained by resubstitution or the hold-out performance is estimated by averaging the results of 1000 experiments. For simplicity, mean A_z will be referred to as A_z in the following discussion. Table I summarizes the combinations with the highest hold-out performance under the various experimental conditions. The various experimental conditions are described in more details in the following.

III.A. Equal covariance matrices with unequal means

In Fig. 1, the three feature selection methods are compared for the LDA classifier. While we do not show the comparisons for every combination investigated, they are discussed in the next section. The A_z values for the resubstitution and the hold-out methods are plotted as a function of $1/N_{\text{train}}$ for $M=50$, 100, and 200. Figures 2 and 3 show the corresponding results for the SVM classifiers with radial and polynomial kernels, respectively. For all three feature selection methods (SFS, SFFS, and PCA), the hold-out A_z for any number of training samples decreased as the number of features available M increased. Examples of the standard deviation values when the SFS, SFFS, and PCA feature selection methods are used with the LDA classifier for $M=100$ are shown in the first row of Fig. 4. The standard deviations of the SVM classifiers (not shown) had similar magnitudes.

Comparing the first and last columns of Fig. 1, the use of PCA for feature selection for LDA appeared to be better than use of SFS under most conditions studied if the number of PCA components was properly chosen. This advantage diminished when $M=200$ and the training sample size was large. In comparison, PCA is an unsupervised feature selection method. LDA with PCA achieved slightly higher hold-out performance than SVM(rad) with PCA when the training sample size was small (compare Figs. 1 and 2). Using SVM(rad) with PCA had a slight advantage for small training sample size and small M ($M=50$) compared with SFS and SFFS, but the hold-out performance with PCA was poorer for $M=100$ or $M=200$.

Figure 3 shows the comparison of feature selection methods with the SVM(poly) classifier. The hold-out A_z of the classifier with SFS at $M=200$ decreased rapidly as F_{in} decreased in the range of large number of training samples, but that was not seen with SFFS. This can be attributed to the fact that the number of features selected by SFS could be larger than 60 for small F_{in}, which was greater than the range of the number of features set to be chosen for the SFFS. At $F_{\text{in}} \geq 5$, where the number of selected features was similar to that of the SFFS, the hold-out A_z was comparable to those for SFFS. SVM(poly) with PCA had the lowest hold-out perfor-

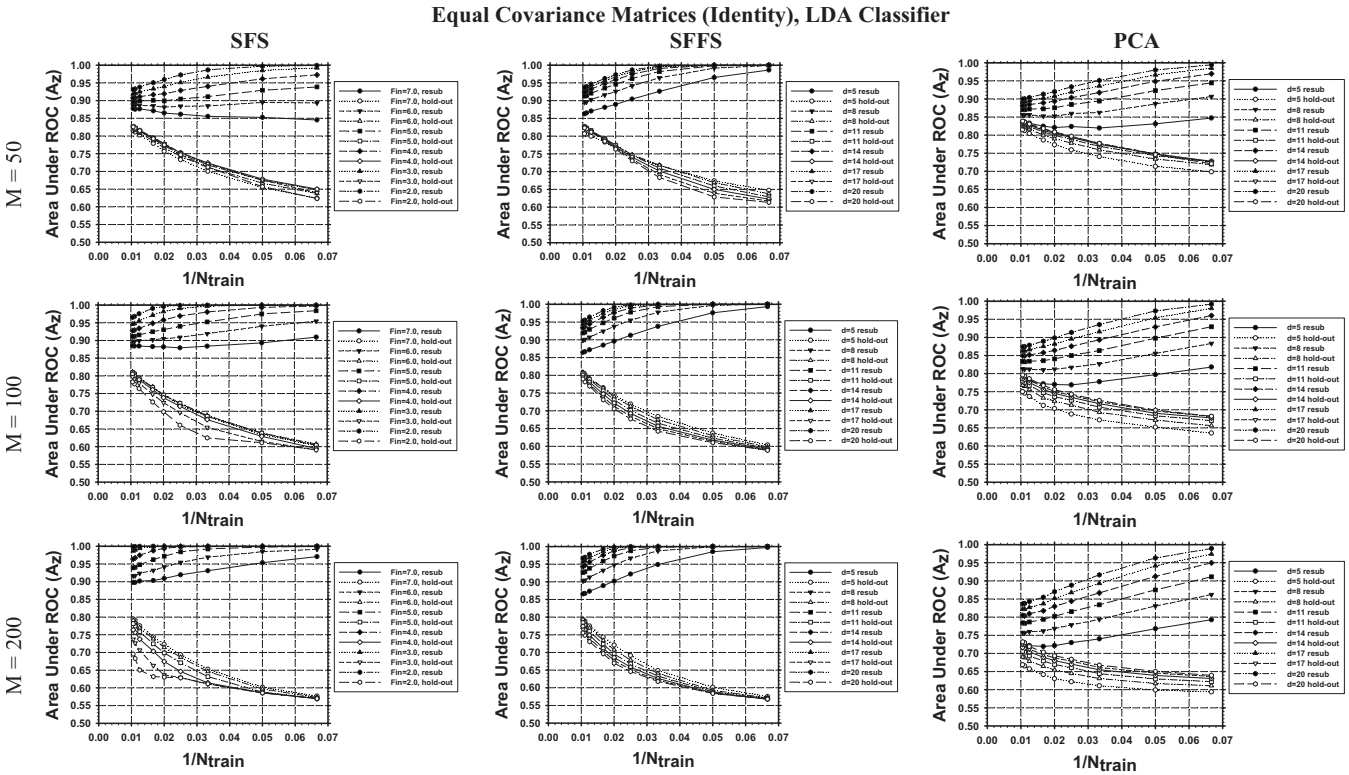

FIG. 1. Dependence of the LDA classifier performance A_z on training sample size. The two class distributions were multivariate normal with equal covariance matrices and unequal means. The effect of increasing dimensionality of the feature space available for selection (M) is shown in each column. The comparison of the SFS, SFFS, and PCA methods for feature selection is shown in each row.

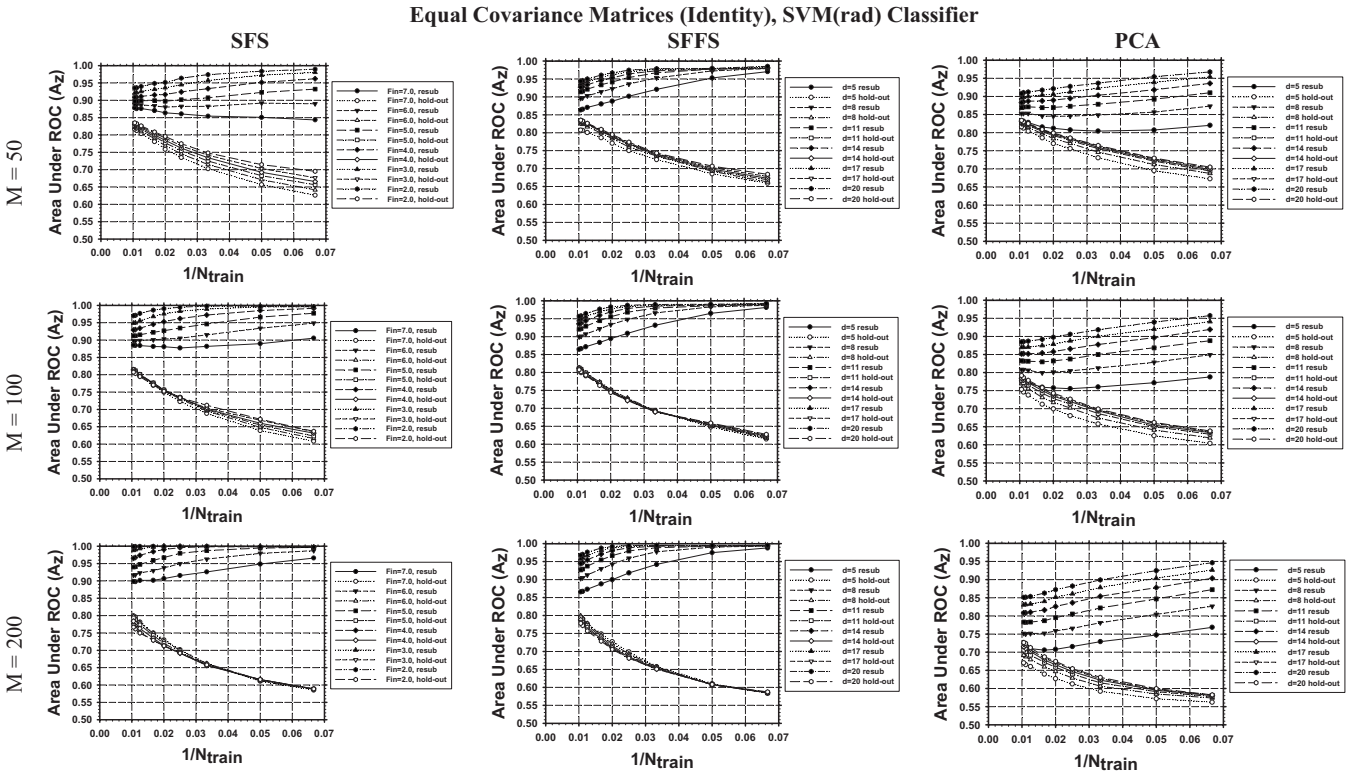

FIG. 2. Dependence of the performance A_z of the SVM classifier with radial kernel on training sample size. The two class distributions were multivariate normal with equal covariance matrices and unequal means. The effect of increasing dimensionality of the feature space available for selection (M) is shown in each column. The comparison of the SFS, SFFS, and PCA methods for feature selection is shown in each row.

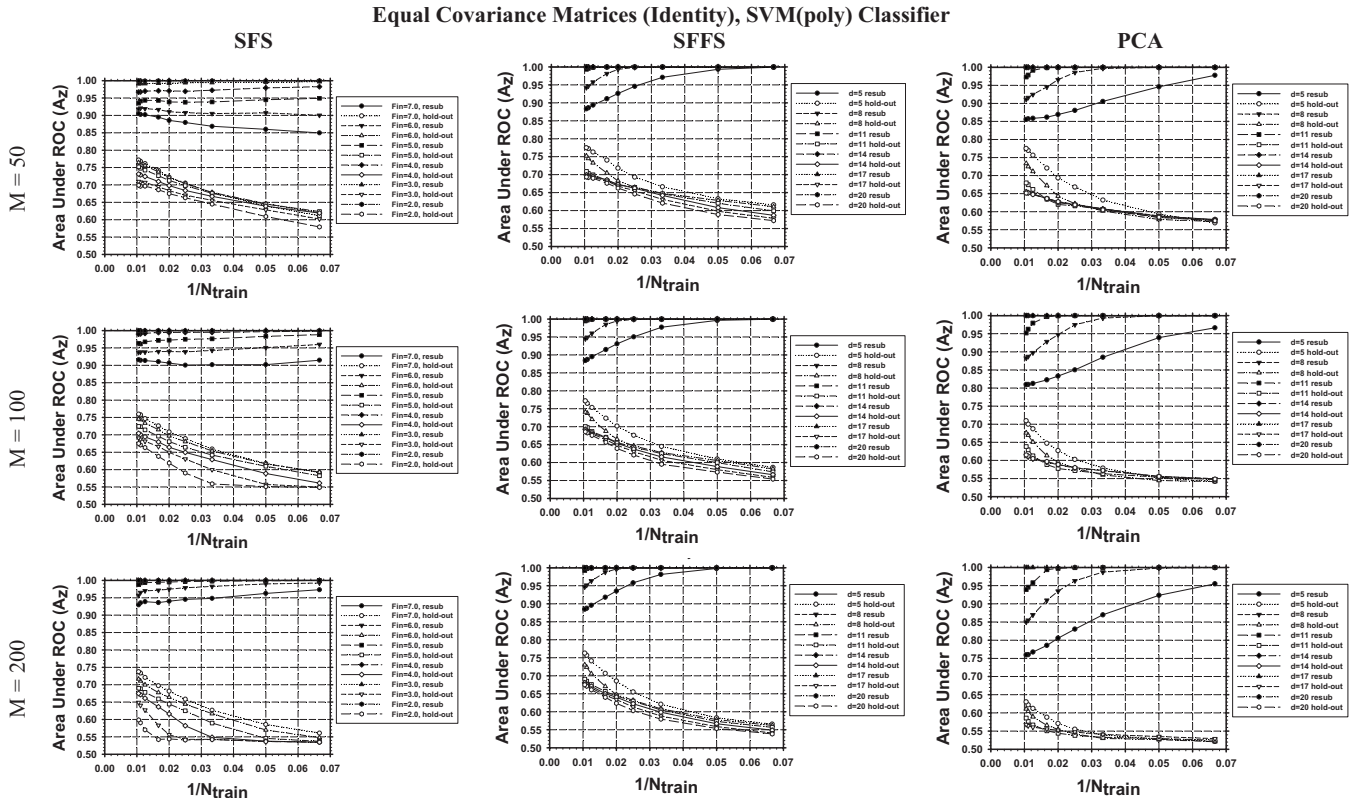

FIG. 3. Dependence of the performance A_z of the SVM classifier with polynomial kernel on training sample size. The two class distributions were multivariate normal with equal covariance matrices and unequal means. The effect of increasing dimensionality of the feature space available for selection (M) is shown in each column. The comparison of the SFS, SFFS, and PCA methods for feature selection is shown in each row.

mance compared to those with SFS and SFFS under the conditions shown except for $M=50$ and large training sample sizes. The hold-out performances of SVM(poly) with PCA were also substantially lower than those of LDA and SVM(rad) with PCA over the parameters studied.

When SFS was used as the feature selection method, the number of selected features increased as M increased for a given F_{in} (graphs not shown). Comparing the first columns of Figs. 1–3, it can be observed that for the LDA, SVM(rad), and SVM(poly) classifiers, the resubstitution A_z increased with increasing M, especially when there were few training samples, whereas the hold-out A_z decreased as a result of overtraining. The hold-out performance for LDA at $M=50$ and for SVM(rad) at $M=50$ and $M=100$ increased as F_{in} decreased, indicating that the additional selected features still increased the discriminatory power under these conditions although likely offset somewhat by the increased dimensionality. The hold-out performance of SVM(poly) was in general worse than those of the LDA and SVM(rad) classifiers.

III.B. Unequal covariance matrices

Comparisons of the feature selection techniques for the SVM(rad) classifier when the two classes had unequal covariance matrices and unequal means are shown in Fig. 5. The results of comparing the classifier performances with input features selected by SFS are shown in Fig. 6. The A_z values of the resubstitution and hold-out estimates of different methods are compared and plotted as a function $1/N_{\text{train}}$. Examples of the standard deviation values when the SFS, SFFS, and PCA feature selection methods are used with the LDA classifier for $M=100$ are shown in the second row of Fig. 4. The standard deviations of the SVM classifiers had similar magnitudes.

The SVM(rad) hold-out performance was virtually the same whether the SFS or SFFS method was used for feature selection when the number of selected features was in the range of five to 20. The hold-out performance with PCA was slightly higher than those with SFS and SFFS for $M=50$ and 100, but the performances with all feature selection methods were comparable for $M=200$. For $M=200$, the number of features selected within the range studied essentially had no influence on hold-out performance.

We have previously shown that LDA may be the preferred classifier when the training sample size is small even under conditions of unequal covariance matrices where it is not the theoretically optimal classifier.[1] A similar comparison of the dependence of the LDA performance on feature selection methods in the same feature space of unequal covariance matrices shows that the dependence of the performance of the LDA classifier on the various study conditions was very similar to that of the SVM(rad) although the A_z values may be different (graphs not shown). The performance of the LDA relative to that of SVM(rad) is demonstrated in Fig. 6 with the SFS method. The SVM(rad) hold-out performance

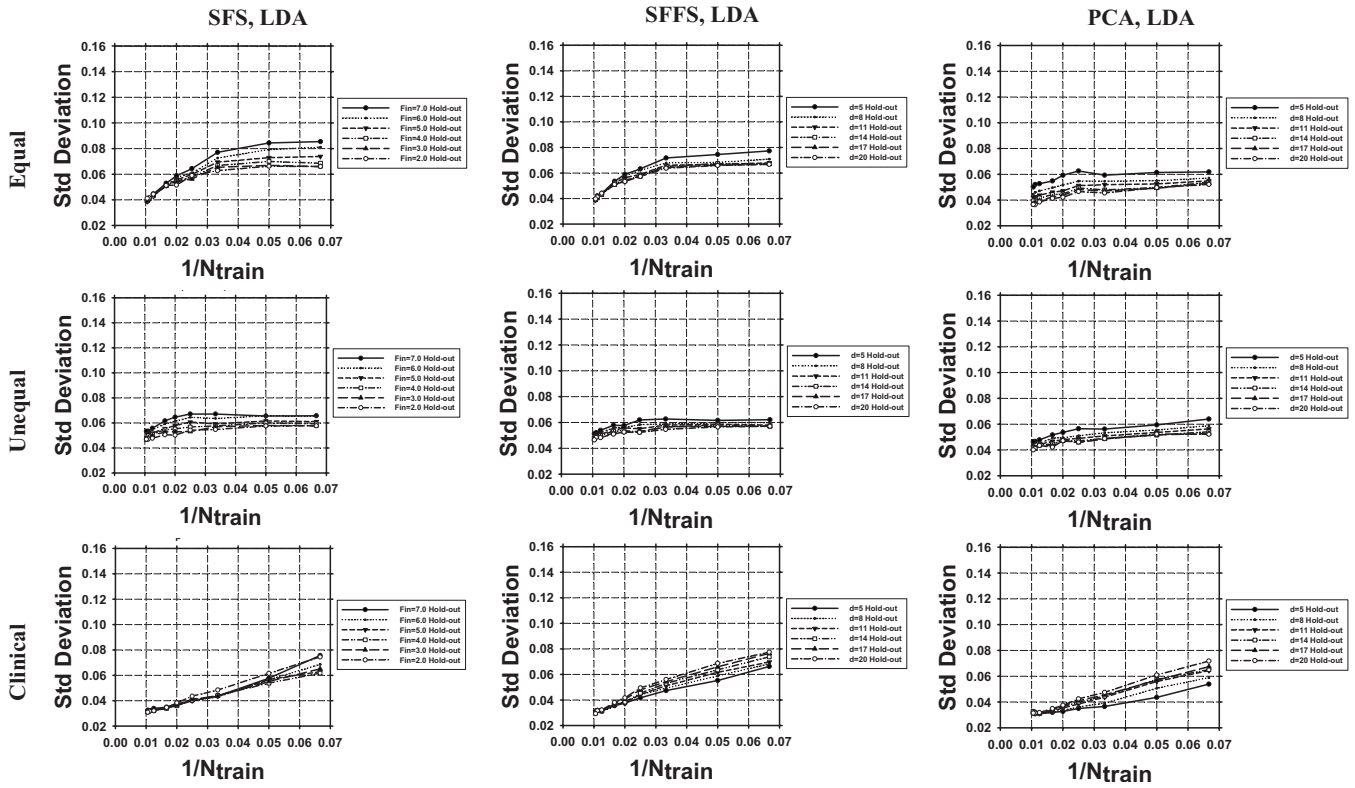

FIG. 4. Standard deviation of the hold-out performance as a function of $1/N_{\text{train}}$ for the SFS, SFFS, and PCA feature selection methods and the LDA classifier. The number of features available for selection was $M=100$ for the equal covariance matrices (first row) and unequal covariance matrices (second row) conditions, and $M=61$ for the condition with simulated equal covariance matrices estimated from a clinical data set.

was comparable or slightly higher than the LDA performance, depending on the number of selected features. With a large number of training samples available, the SVM(rad) hold-out performance was better than that of the LDA, especially when $M=200$ and the number of selected features was large.

When the SFFS method was used for feature selection (graphs not shown), the SVM(poly) classifier had the highest resubstitution bias for small training sample sizes. The hold-out performance for SVM(poly) was similar to those of LDA at small training sample sizes, and was slightly better at large training samples sizes. SVM(rad) consistently had a slightly higher hold-out performance than LDA, especially for $M=50$ with small training sample sizes. Within the range of number of features selected by SFFS (five to 20) in this study, the hold-out performances of LDA and SVM(rad) were almost independent of the number of features used. The latter result can be seen in the middle column of Fig. 5. Comparing the classifiers with PCA (graphs not shown), it is observed that the SVM(poly) classifier suffered stronger bias than the other two classifiers, especially when the dimensionality was high. Its hold-out performance was lower and the resubstitution performance was higher than those of LDA or SVM(rad), similar to those observed for the class distributions with equal covariance matrices. LDA and SVM(rad) performed similarly.

III.C. Equal covariance matrices (clinical)

In these experiments, the two classes had unequal means but the same covariance matrix derived from the features extracted from lung nodules on CT scans. There were $M=61$ features available for selection. The A_z values from the LDA, SVM(rad), and SVM(poly) classifiers with the SFS, SFFS, and PCA feature selection techniques are compared in Fig. 7. Examples of the standard deviation values when the three feature selection methods are used with the LDA classifier are shown in the third row of Fig. 4.

SVM(rad) had less optimistic resubstitution bias and less pessimistic hold-out bias compared to LDA under the conditions of small training sample size. The differences between LDA and SVM(rad) in this feature space were greater than those observed in the class distributions with simulated equal and unequal covariance matrices discussed above. However, when the training sample size approached about 100 samples per class, LDA with SFS and SFFS provided a slightly higher hold-out A_z than SVM(rad). SVM(poly) again had the lowest hold-out performance among the three.

PCA provided slightly higher hold-out performance than those obtained with the SFFS method for LDA when the training sample size was small, but comparable or lower hold-out performances for other conditions.

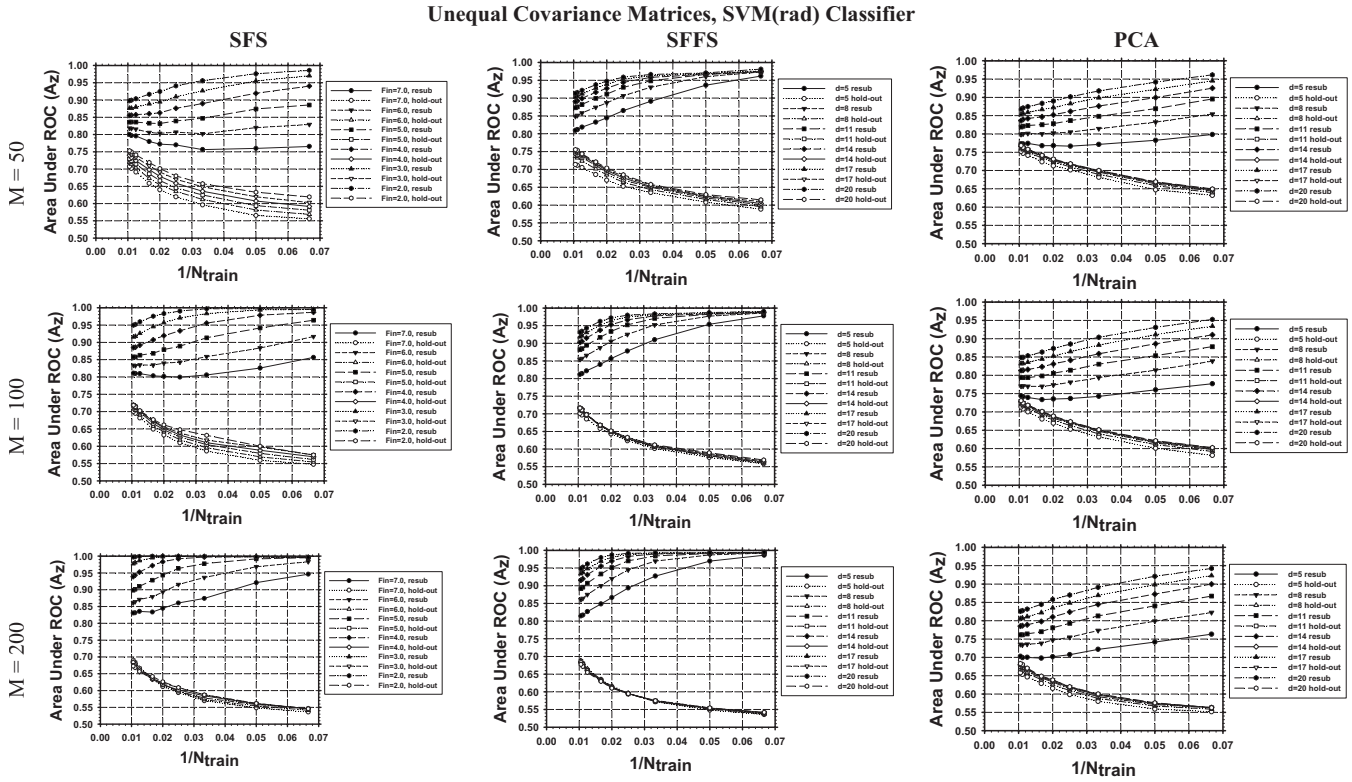

FIG. 5. Dependence of the performance A_z of the SVM classifier with radial kernel on training sample size. The two class distributions were multivariate normal with unequal covariance matrices and unequal means. The effect of increasing dimensionality of the feature space available for selection (M) is shown in each column. The comparison of the SFS, SFFS, and PCA methods for feature selection is shown in each row.

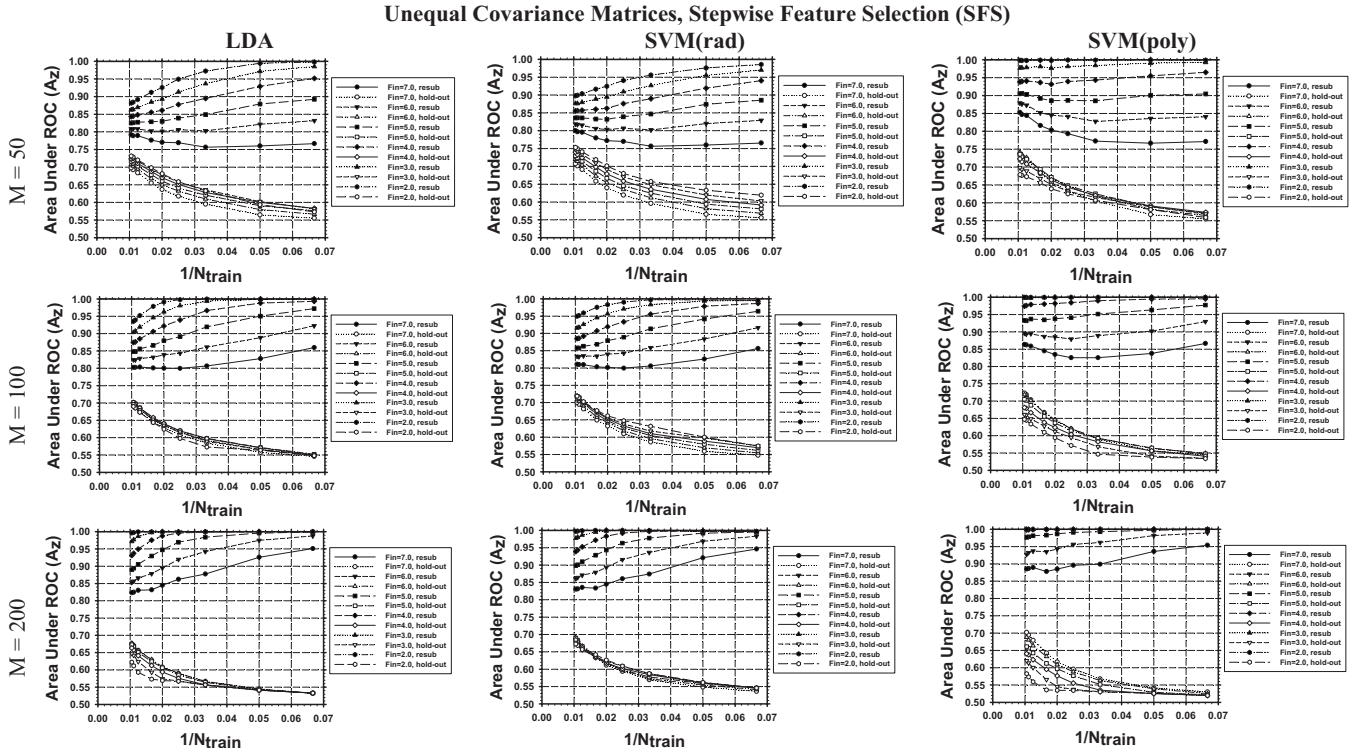

FIG. 6. Comparison of the LDA, SVM(rad), and SVM(poly) classifiers with the same input features obtained from SFS. The two class distributions were multivariate normal with unequal covariance matrices and unequal means.

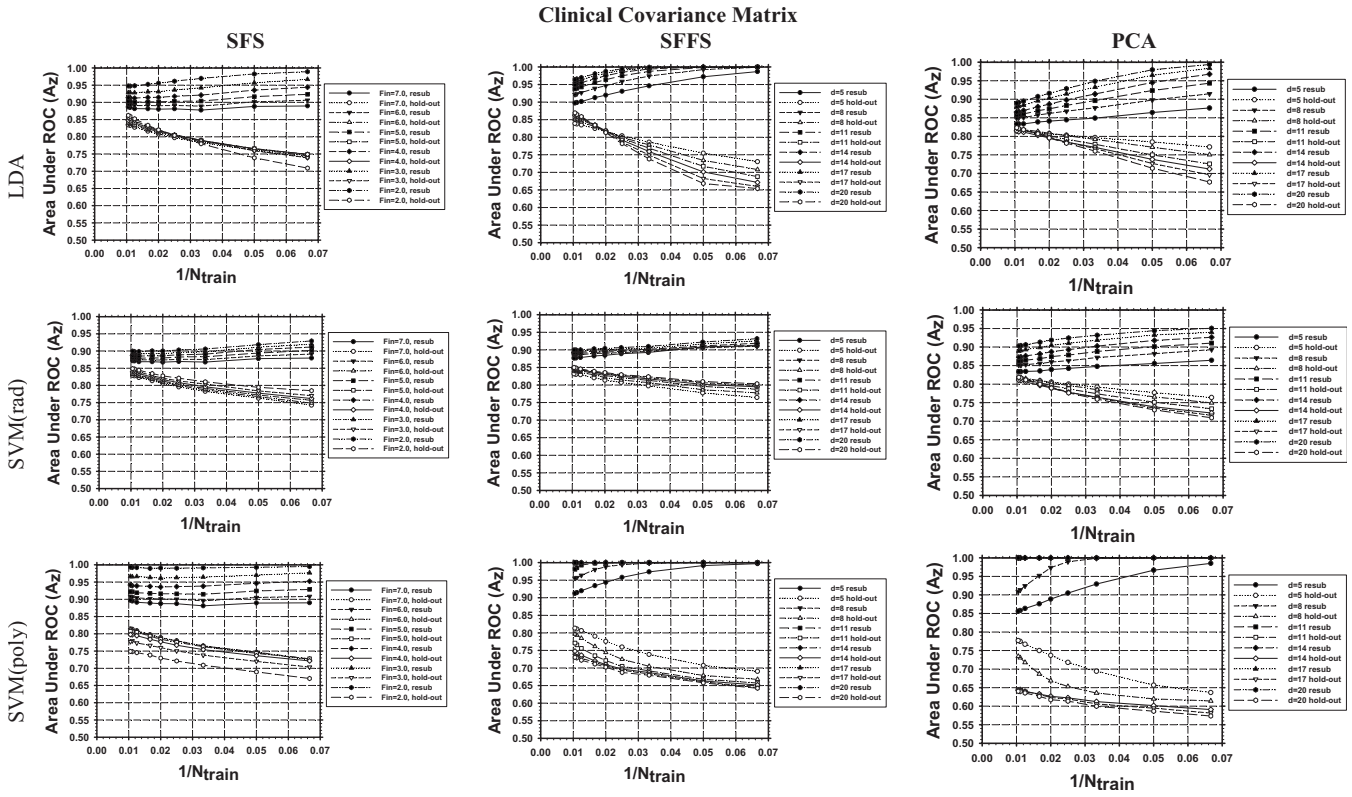

FIG. 7. Performance of the SFS, SFFS, and PCA feature selection methods and the LDA, SVM(rad), and SVM(poly) classifiers for simulated multivariate normal class distributions with equal covariance matrices estimated from a clinical data set ($M=61$).

IV. DISCUSSION

Numerous feature selection and classifier methods have been investigated in the literature. It is difficult to determine which combination of feature selection and classifier methods would be the most effective for a given classification task. When a specific combination is selected based on a limited number of samples available, as often is the case for CAD system development, the potential for overtraining is high, and the performance of the resulting CAD system as predicted by the training data may not be generalizable to the population at large. To demonstrate the effects of different combinations, we conducted a simulation study with sample data randomly drawn from multivariate Gaussian distributions, which allowed us to generate an arbitrarily large number of samples.

Although we investigated both resubstitution and hold-out methods and compared the results obtained from both methods in all the graphs, the trends of the hold-out performance are emphasized more than those of the resubstitution performance in our discussion. The hold-out performance of classifiers is more important than their resubstitution performance because it is generally accepted that the resubstitution performance is optimistically biased and should not be used in ultimate CAD system evaluation. When the number of training samples is small, the classifier is more easily overtrained, and this overtraining usually results in an optimistic predicted performance when the classifier is applied to the same data set that has been used for training, compared to what would be obtained if the trained classifier were applied to the true population. The real test of whether a CAD system is effective should be performed with samples it has not seen before, which is modeled by the hold-out method.

A limitation to this study was that we investigated only a small number of combinations, and the results may not be directly applicable to features extracted from clinical data for which the class distributions may not be Gaussian. Although we expect that the hold-out method will have a pessimistic bias (relative to what can be achieved if the classifier is trained with an infinite sample size) regardless of whether Gaussian or non-Gaussian data are used, the relative trend of the bias may be different if samples drawn from clearly non-Gaussian distributions are used. Given that there are large numbers of CAD applications and a wide range of features that can be extracted in each application, the class distributions can be extremely varied. It will not be possible to cover even a small fraction of the distributions that may occur in clinical data in one study. Nevertheless, through a systematic investigation of the commonly assumed class distributions (Gaussians with equal or unequal covariance matrices), the trends observed will lead to better understanding of some characteristics of the factors in classifier design, which may serve as a guide in future developments.

When our results were examined, some unifying themes that held for both the equal and unequal covariance matrix conditions emerged. These themes are described next, fol-

lowed by specific observations for the three conditions: Equal covariance matrices, unequal covariance matrices, and equal covariance matrices (clinical).

In Figs. 1–3, 5, and 6, the effect of increasing the feature space dimensionality is demonstrated by comparing the graphs along each column. For equal covariances, the contribution of additional features beyond $i=25$ was close to zero because the ability of feature i to separate the two classes decreased with increasing i, the feature index, in Eq. (2). For unequal covariances, the values of v_i in Eq. (3) can be approximated by $v_i=1+(v_{\max}-1)(\gamma^{1-i}-\gamma^{1-M})$ for large γ^M. With the values of γ and M used in our experiments ($\gamma=1.5$ and $M=50$, 100, or 200), the contribution of additional features beyond $i=25$ was also close to zero for the unequal covariance condition. An effective feature selection algorithm should be impervious to these additional features, which were essentially noise since the difference in the means or covariances of a given feature between the two classes from which samples were drawn was close to zero. The decrease in hold-out A_z with the increase in the number of features available M indicates that the presence of useless features can interfere with the selection of the good features when the sample size is small.

When a large dimensional feature space is available for feature selection, it can be expected that the hold-out performance would increase initially as the effective feature combinations are being found. However, when more and more features are added, the hold-out A_z would eventually decrease due to overtraining on the finite design sample size, a phenomenon known as the curse of dimensionality in the literature.[8] The number of selected features, or the F_{in}, F_{out} thresholds, for which the hold-out performance reaches a maximum will depend on the properties of the feature space and classifier used, as seen from the examples of various conditions included in this study. For example, for LDA with SFS and equal covariance matrices, this peak occurs at different F_{in} values for different values of M and number of training samples (Fig. 1). Interestingly, for most of the conditions shown in Fig. 2 where the performance of SVM(rad) with equal covariance matrices is assessed, the hold-out performance increased uniformly or stayed almost unchanged when the F_{in} values decreased, or the number of selected features increased. The different trend of dependence on the dimensionality of the selected feature space for the classifiers can be observed more readily in Figs. 5 and 6 for the unequal covariance matrices. For example, in Fig. 6 with $M=100$ (middle row), $F_{\text{in}}=2$ had worse performance compared to larger F_{in} values with LDA and SVM(poly), while it had the best performance with SVM(rad). Since all classification methods used the same selected features, and lower F_{in} steers the SFS to select a larger number of features, this provides evidence that SVM(rad) classifier may be more immune to overtraining with a large number of features than LDA. This is in agreement with previous observations in the literature that SVM is capable of circumventing the curse of dimensionality.[34] However, our study also indicates that this capability has its limits. The last row of Fig. 2 shows that there exist conditions for which $F_{\text{in}}=2$ results in a worse performance compared to larger F_{in} values with SVM(rad). We observed that for different number of training samples, the number of selected features was mostly larger than 60 under this condition.

Our study also highlighted the importance of kernel selection in SVM. Different kernels introduce different nonlinearities into the mapping between the original feature space and the higher-dimensional Euclidean space H. It has been noted in the literature that kernel selection can have an important impact on the performance of the SVMs.[39] There are a large number of options for the SVM kernel. We implemented only two popular kernels. Our results indicated that within the conditions evaluated (Figs. 3 and 6), the hold-out performance of SVM(poly) decreased uniformly when the number of selected features increased by any feature selection methods, opposite to the trend of SVM(rad). SVM(poly), therefore, appeared to be more vulnerable to overtraining with a large number of input features than SVM(rad). Overall, the radial kernel had a small but consistent advantage over the polynomial kernel under most conditions, and did not have a disadvantage under any of the conditions that were included in our simulation study.

Under the conditions studied, the resubstitution A_z for SFS at a given sample size increased as the F_{in} and F_{out} thresholds decreased, i.e., as the feature selection method was steered to select a larger number of features. Likewise, the resubstitution A_z of SFFS increased with an increased number of features selected. At small sample sizes, SFS had a lower resubstitution A_z than SFFS, especially at low M and high F_{in} (e.g., comparing $F_{\text{in}}=7.0$ to $d=5$ at $M=50$ in Figs. 1 and 5). This may be partially attributed to the small number of features (<5) selected by SFS under these conditions, and partially to the fact that SFFS may be able to select features that fit the training data better. However, comparing the hold-out A_z values for the same conditions, SFFS did not suffer a noticeable tradeoff for its higher resubstitution bias. Conversely, for large M and low F_{in} (e.g., comparing $F_{\text{in}}=2.0$ to $d=20$ at $M=200$ in Figs. 1 and 5), SFS had a higher resubstitution A_z than SFFS, likely due to the large number of features selected by SFS. For LDA, this overfitting of the training data by SFS resulted in lower hold-out A_z values than for SFFS (e.g., comparing $F_{\text{in}}=2.0$ to $d=20$ at $M=200$ in Fig. 1). For SVM(rad), the hold-out A_z values were almost identical for the entire F_{in} range at $M=200$, indicating again that SVM(rad) may be more tolerant than LDA to conditions that might lead to overfitting.

IV.A. Equal covariance matrices with unequal means

For the LDA, PCA performed better with a higher hold-out performance than SFS except when M and training sample size were large. This is somewhat unexpected because the Wilks' lambda condition used in this study for feature selection was closely matched to a linear model. It is interesting to note that given the same selected features from the SFS method, the SVM(rad) had slightly better hold-out performance than the LDA, especially when the number of

available features M and the number of selected features were large. Since the data were drawn from multivariate normal distributions with identical covariance matrices, it is expected that LDA would theoretically provide the optimal performance. However, LDA estimated the means and covariance matrices from the available training samples of both classes, and the limited sample size could result in poor estimates. The difference in the performance between LDA and SVM(rad) decreased as the training sample size increased. When the number of training samples reached the highest for the experimental conditions studied (100 per class), the LDA hold-out performance was similar to the SVM(rad) performance when the number of selected features was small. For the SFS method and a given training sample size, the hold-out A_z for SVM(rad) was less dependent on the F_{in} and F_{out} thresholds than that of LDA when the original feature space dimensionality was high ($M=100$ or 200) and vice versa when the dimensionality was low ($M=50$). Given conditions similar to this study, SVM(rad) would have a slight advantage over LDA for large dimension feature spaces.

When the SFFS method was used for feature selection (central columns of Figs. 1–3), it can be observed that the general trends of the three classifiers were similar to those using the SFS method. In the range of five to 20 selected features, the hold-out performance for all three classifiers did not have a strong dependence on the number of selected features.

IV.B. Unequal covariance matrices

The SVM(rad) hold-out performance was similar for both SFS and SFFS while that with PCA was slightly higher at small training sample size. For all feature selection methods at $M=200$, the hold-out performance was similar. Given the design of the features, the additional features did not provide much discriminatory power, and the SVM(rad) classifier may have effectively disregarded them. Since the SVM(rad) classifier is nonlinear, it would be expected to perform better than LDA in the feature spaces with unequal covariance matrices. A marginally higher performance for SVM(rad) was observed when SFS and SFFS were used. The combination of LDA with PCA improved the performance of LDA to the level of SVM(rad) for $M=50$ and 100 and slightly higher than SVM(rad) for $M=100$ and 200 at small sample size (graphs not shown).

IV.C. Equal covariance matrices (clinical)

For this class distribution, neither LDA nor SVM(rad) with PCA seems to have an advantage over the SFS and SFFS methods as observed for the two previous class distributions for a given M (e.g., $M=50$). However, there was a larger difference between LDA and SVM(rad) performance compared to the samples drawn from the simulated equal or unequal covariance matrices described above. When the training sample size was large, LDA with SFS and SFFS provided a slightly higher hold-out A_z than SVM(rad) but vice versa when the sample size was small.

V. CONCLUSION

The LDA classifier has been used for many classification tasks in CAD applications because of the limited number of samples available for training and testing. A linear classifier would less likely overfit the training data because of the relatively few parameters to be trained. Recently, there has been increased interest in the SVM. Under our simulation conditions, we found that the SVM with the radial kernel performed slightly better than the LDA when the training sample size was small for the data drawn from the covariance matrices estimated from clinical data. Under the other simulated conditions in our study, PCA with LDA was effective for small training sample sizes. However, the performance of SVM for a specific classification task depends on many variables that need to be selected, such as the kernel function and the parameter values. A different choice of kernel, such as the polynomial function in this simulation study, may result in lower performance than the LDA under many of the conditions. Although we could only examine a limited number of conditions in the current study, we demonstrated that the relative performances of the different combinations of classifier and feature selection method depend on the feature space distributions, the dimensionality, and the available training sample sizes. Further investigations will be needed to determine if there can be simple rules of thumb to guide the choice among different classifiers, or among the kernel functions for SVM. From the comparison of feature selection methods, we found that the SFS and the SFFS methods are similar while PCA can provide higher hold-out performance than SFS and SFFS under some conditions.

Choosing effective feature selection and classification methods is a vital part in the development of a CAD system. Our study has revealed some interesting properties of these methods. The knowledge of the interaction between the feature selection and classification methods may facilitate the design of an effective CAD system under the constraint of limited available samples.

ACKNOWLEDGMENTS

This work is supported by USPHS Grant Nos. CA95153 and CA 93517.

[a]Author to whom correspondence should be addressed. Electronic mail: berki@umich.edu; Telephone: 734-647-7429; Fax: 734-615-5513.

[1]H. P. Chan, B. Sahiner, R. F. Wagner, and N. Petrick, "Classifier design for computer-aided diagnosis: Effects of finite sample size on the mean performance of classical and neural network classifiers," Med. Phys. **26**, 2654–2668 (1999).
[2]B. Sahiner, H. P. Chan, N. Petrick, R. F. Wagner, and L. M. Hadjiiski, "Feature selection and classifier performance in computer-aided diagnosis: The effect of finite sample size," Med. Phys. **27**, 1509–1522 (2000).
[3]B. Sahiner, H. P. Chan, and L. Hadjiiski, "Classifier performance prediction for computer-aided diagnosis using a limited data set," Med. Phys. **35**, 1559–1570 (2008).
[4]B. Sahiner, H. P. Chan, and L. M. Hadjiiski, "Classifier performance estimation under the constraint of a finite sample size: Resampling schemes applied to neural network classifiers," Neural Networks **21**, 476–483 (2008).
[5]Q. Li and K. Doi, "Analysis and minimization of overtraining effect in rule-based classifiers for computer-aided diagnosis," Med. Phys. **33**, 320–328 (2006).

[6]Q. Li and K. Doi, "Comparison of typical evaluation methods for computer-aided diagnostic schemes: Monte Carlo simulation study," Med. Phys. **34**, 871–876 (2007).
[7]S. V. Beiden, M. A. Maloof, and R. F. Wagner, "A general model for finite-sample effects in training and testing of competing classifiers," IEEE Trans. Pattern Anal. Mach. Intell. **25**, 1561–1569 (2003).
[8]A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample size performance," IEEE Trans. Pattern Anal. Mach. Intell. **19**, 153–158 (1997).
[9]P. Pudil, J. Novovicová, and J. Kittler, "Floating search methods in feature selection," Pattern Recogn. Lett. **15**, 1119–1125 (1994).
[10]M. Kudo and J. Sklansky, "Comparison of algorithms that select features for pattern classifiers," Pattern Recogn. **33**, 25–41 (2000).
[11]C. Sima and E. R. Dougherty, "What should be expected from feature selection in small-sample settings," Bioinformatics **22**, 2430–2436 (2006).
[12]J. P. Hua, W. D. Tembe, and E. R. Dougherty, "Performance of feature-selection methods in the classification of high-dimension data," Pattern Recogn. **42**, 409–424 (2009).
[13]J. W. Lee, J. B. Lee, M. Park, and S. H. Song, "An extensive comparison of recent classification tools applied to microarray data," Comput. Stat. Data Anal. **48**, 869–885 (2005).
[14]X. G. Zhang, X. Lu, Q. Shi, X. Q. Xu, H. C. E. Leung, L. N. Harris, J. D. Iglehart, A. Miron, J. S. Liu, and W. H. Wong, "Recursive SVM feature selection and sample classification for mass-spectrometry and microarray data," BMC Bioinf. **7**, 13 (2006).
[15]J. G. Dy and C. E. Brodley, "Feature selection for unsupervised learning," J. Mach. Learn. Res. **5**, 845–889 (2004).
[16]S. Krishnan, K. Samudravijaya, and P. V. S. Rao, "Feature selection for pattern classification with Gaussian mixture models: A new objective criterion," Pattern Recogn. Lett. **17**, 803–809 (1996).
[17]R. Kohavi and G. H. John, "Wrappers for feature subset selection," Artif. Intell. **97**, 273–324 (1997).
[18]L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," J. Mach. Learn. Res. **5**, 1205–1224 (2004).
[19]K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. (Academic, New York, 1990).
[20]R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
[21]D. J. Hand, *Discrimination and Classification* (Wiley, New York, 1981).
[22]H. H. Barrett, C. K. Abbey, and E. Clarkson, "Objective assessment of image quality. III. ROC metrics, ideal observers, and likelihood-generating functions," J. Opt. Soc. Am. A Opt. Image Sci. Vis. **15**, 1520–1535 (1998).
[23]T. W. Way, L. M. Hadjiiski, B. Sahiner, H.-P. Chan, P. N. Cascade, E. A. Kazerooni, N. Bogot, and C. Zhou, "Computer-aided diagnosis of pulmonary nodules on CT scans: Segmentation and classification using 3D active contours," Med. Phys. **33**, 2323–2337 (2006).
[24]M. M. Galloway, "Texture classification using gray level run lengths," Comput. Graph. Image Process. **4**, 172–179 (1975).
[25]B. R. Dasarathy and E. B. Holder, "Image characterizations based on joint gray-level run-length distributions," Pattern Recogn. Lett. **12**, 497–502 (1991).
[26]T. Marill and D. Green, "On the effectiveness of receptors in recognition systems," IEEE Trans. Inf. Theory **9**, 11–17 (1963).
[27]A. W. Whitney, "A direct method of nonparametric measurement selection," IEEE Trans. Comput. **C-20**, 1100–1103 (1971).
[28]N. R. Draper, *Applied Regression Analysis* (Wiley, New York, 1998).
[29]M. M. Tatsuoka, *Multivariate Analysis, Techniques for Educational and Psychological Research*, 2nd ed. (Macmillan, New York, 1988).
[30]M. J. Norusis, *SPSS for Windows Release 6 Professional Statistics* (SPSS, Chicago, 1993).
[31]S. D. Stearns, "On selecting features for pattern classifiers," Third International Conference on Pattern Recognition, Coronado, CA, 1976, pp. 71–75.
[32]A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," IEEE Trans. Pattern Anal. Mach. Intell. **22**, 4–37 (2000).
[33]P. A. Lachenbruch, *Discriminant Analysis* (Hafner, New York, 1975).
[34]C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," Data Min. Knowl. Discov. **2**, 121–167 (1998).
[35]Y. Arzhaeva, M. Prokop, D. M. J. Tax, P. A. De Jong, C. M. Schaefer-Prokop, and B. van Ginneken, "Computer-aided detection of interstitial abnormalities in chest radiographs using a reference standard based on computed tomography," Med. Phys. **34**, 4798–4809 (2007).
[36]P. Campadelli, E. Casiraghi, and D. Artioli, "A fully automated method for lung nodule detection from postero-anterior chest radiographs," IEEE Trans. Med. Imaging **25**, 1588–1603 (2006).
[37]A. K. Jerebko, J. D. Malley, M. Franaszek, and R. M. Summers, "Support vector machines committee classification method for computer-aided polyp detection in CT colonography," Acad. Radiol. **12**, 479–486 (2005).
[38]S. Ruping, "Incremental learning with support vector machines," Proceedings of the IEEE International Conference on Data Mining, 2001, pp. 641–642.
[39]O. Chapelle, P. Haffner, and V. N. Vapnik, "Support vector machines for histogram-based image classification," IEEE Trans. Neural Netw. **10**, 1055–1064 (1999).