

Feature Selection for Gene Expression Using Model-Based Entropy

Shenghuo Zhu, Dingding Wang, Kai Yu, Tao Li, and Yihong Gong

Abstract—Gene expression data usually contain a large number of genes but a small number of samples. Feature selection for gene expression data aims at finding a set of genes that best discriminate biological samples of different types. Using machine learning techniques, traditional gene selection based on empirical mutual information suffers the data sparseness issue due to the small number of samples. To overcome the sparseness issue, we propose a model-based approach to estimate the entropy of class variables on the model, instead of on the data themselves. Here, we use multivariate normal distributions to fit the data, because multivariate normal distributions have maximum entropy among all real-valued distributions with a specified mean and standard deviation and are widely used to approximate various distributions. Given that the data follow a multivariate normal distribution, since the conditional distribution of class variables given the selected features is a normal distribution, its entropy can be computed with the log-determinant of its covariance matrix. Because of the large number of genes, the computation of all possible log-determinants is not efficient. We propose several algorithms to largely reduce the computational cost. The experiments on seven gene data sets and the comparison with other five approaches show the accuracy of the multivariate Gaussian generative model for feature selection, and the efficiency of our algorithms.

Index Terms—Feature selection, multivariate Gaussian generative model, entropy.

1 INTRODUCTION

GENE expression refers to the level of production of protein molecules defined by a gene. Monitoring of gene expression is one of the most fundamental approach in genetics and molecular biology. The standard technique for measuring gene expression is to measure the mRNA instead of proteins, because mRNA sequences hybridize with their complementary RNA or DNA sequences while this property lacks in proteins. The DNA arrays, pioneered in [5] and [10], are novel technologies that are designed to measure gene expression of tens of thousands of genes in a single experiment. The ability of measuring gene expression for a very large number of genes, covering the entire genome for some small organisms, raises the issue of characterizing cells in terms of gene expression, that is, using gene expression to determine the fate and functions of the cells. The most fundamental of the characterization problem is that of identifying a set of genes and its expression patterns that either characterize a certain cell state or predict a certain cell state in the future [18].

When the expression data set contains multiple classes, the problem of classifying samples according to their gene expression becomes much more challenging, especially when the number of classes exceeds five [24]. Moreover, the special characteristics of expression data add more

challenge to the classification problem. Expression data usually contain a large number of genes (in thousands) and a small number of experiments (in dozens). In machine learning terminology, these data sets are usually of very high dimensions with undersized samples. In microarray data analysis, many gene selection methods have been proposed to reduce the data dimensionality [31].

Gene selection aims to find a set of genes that best discriminate biological samples of different types. The selected genes are “biomarkers,” and they form a “marker panel” for analysis. Most gene selection schemes are based on binary discrimination using rank-based schemes [8] such as information gain, which reduces the entropy of the class variables given the selected features. One critical issue in these rank-based methods is data sparseness. For example, the estimation of the traditional information gain is an empirical estimation directly on the data. Suppose we select the 11th gene for a data set. The 10 selected genes split the training data into $1,024 = 2^{10}$ groups (assuming that each gene does a binary split). Since we have very few samples in most groups, the estimations of mutual information between the 11th gene and the target in each group are not accurate. Thus, the information gain, which is the sum of the mutual information over all groups, is not accurate.

To overcome the issue of data sparseness, we propose a model-based approach to estimate the entropy on the model, instead of on the data themselves. Here, we use multivariate Gaussian generative models, which model the data with multivariate normal distributions. Multivariate normal distributions are widely used in various areas, including gene expression data [34], because of their *generality* and *simplicity*. The means of variables (expression data of genes and class labels) and the covariances between them are two basic measures of variables themselves and

• S. Zhu, K. Yu, and Y. Gong are with NEC Laboratories America, 10080 North Wolfe Road, Suite SW3-350, Cupertino, CA 95014. E-mail: {zsh, kyu, ygong}@sv.nec-labs.com.

• D. Wang and T. Li are with the School of Computer Science, Florida International University, 11200 SW 8th Street, Miami, FL 33199. E-mail: {dwang003, taoli}@cs.fiu.edu.

Manuscript received 18 Oct. 2007; revised 17 Jan. 2008; accepted 13 Mar. 2008; published online 10 Apr. 2008.

For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-2007-10-0136. Digital Object Identifier no. 10.1109/TCBB.2008.35.

the interaction between them. To predict the classes of data, we have to model the interaction between genes and class labels. Given the mean and covariance, multivariate Gaussian is the distribution with the maximum entropy, which implies its *generality* according to the principle of maximum entropy [13]. Usually, we can explicitly and efficiently estimate parameters of multivariate Gaussian models via a few matrix operations, which implies the *simplicity* of the models. Though the class variables are binary or categorical, we relax them as numerical values, which bring us the simplicity. Our experiments suggest that this approximation in the feature selection does not affect the classification accuracy.

A nice property of multivariate Gaussian distributions is that the conditional distribution of a subset of variables given another subset of variables is still a multivariate Gaussian distribution. We can explicitly compute the entropy of class variables given the selected features with the log-determinant of the covariance matrix of the conditional distribution [6], [3]. Therefore, the objective of minimizing the entropy becomes to find a set of features to minimize the log-determinant of the conditional covariance matrix. Because of the large number of genes, the computation of all log-determinants of the conditional covariance matrix is not time consuming. We propose several algorithms to largely reduce the computational cost.

In summary, our contributions are the following:

1. We propose a model-based approach to estimate the entropy based on multivariate normal distributions. The model-based approach addresses the data sparseness problem in gene selection.
2. We propose several algorithms to efficiently compute all log-determinants of the conditional covariance matrix and largely reduce the computational cost. The assumption of multivariate Gaussian generative models leads to simple, robust, and effective computation methods for gene selection.
3. We perform extensive experimental study on seven gene data sets and compare our algorithms with other five approaches.

The rest of the paper is organized as follows: A brief note on the related work is given in Section 2. The notation used in this paper is listed in Section 3. Our algorithms are presented in Section 4, and the comparison methodologies are described in Section 5. We show the experimental results in Section 6 and discuss the idea of experimental designs and its connection with gene selection in Section 7. Finally, Section 8 concludes.

2 RELATED WORK

Generally two types of feature selection methods have been studied in the literature: filter methods [17] and wrapper methods [16]. Filter-type methods are essentially data preprocessing or data filtering methods. Features are selected based on the intrinsic characteristics that determine their relevance or discriminative powers with regard to the target classes. In wrapper-type methods, feature selection is

“wrapped” around a learning method: the usefulness of a feature is directly judged by the estimated accuracy of the learning method. Wrapper methods typically require extensive computation to search for the best features. As pointed out in [32], the essential differences between the two methods are

1. that a wrapper method makes use of the algorithm that will be used to build the final classifier while a filter method does not, and
2. that a wrapper method uses cross validation to compare the performance of the final classifier and searches for an optimal subset while a filter method uses simple statistics computed from the empirical distribution to select an attribute subset.

Wrapper methods could perform better but would require much more computational costs than filter methods. Most gene selection schemes are based on binary discrimination using rank-based filter methods [8] such as t-statistics and information gain [31]. One critical issue in these rank-based methods is data sparseness, and most of the rank-based methods do not take redundancy into consideration. In order to remove the redundancy among features, a Min-Redundancy and Max-Relevance (mRMR) framework is proposed in [25]. Yu and Liu [36] also explored the relationship between feature relevance and redundancy and proposed a method that can effectively remove redundant genes. In addition, ReliefF, a widely used feature subset selection method, has also been applied to gene selection [22]. In the paper [19], a hierarchical Bayesian model is used to enforce the sparsity in the features. However, the sparsity is indirectly controlled by a parameter, not specified by the given number of features to select.

In Section 5, we will describe several gene selection methods used in our experimental comparisons in detail. In this paper, we propose a model-based approach to estimate the information gain, instead of on the data itself. This would overcome the limitations of data sparseness. In addition, the model parameters can be explicitly and efficiently estimated via a few matrix operations.

3 NOTATION

A summary of the notation we use in this paper is shown in Table 1.

4 MODEL-BASED FEATURE SELECTION

4.1 Feature Selection Using Entropy Measure

Suppose we have f feature variables of the underlying data, denoted by $\{X_i | i \in F\}$, where F is the full feature index set, having $|F| = f$. We have the class variable, Y , represented by multiple class indicator variables. For example, in a three-class classification problem, the class variables are represented by vectors $(1, -1, -1)$, $(-1, 1, -1)$, and $(-1, -1, 1)$. The problem of feature selection is to select a subset of features, $S \subset F$, to accurately predict the target Y , given that the cardinality of S is m ($m < f$). Let us denote $\{X_i | i \in S\}$ by X_S , for any set S .

TABLE 1
A Summary of Notation

\mathbf{K}	a matrix
\mathbf{K}_{SR}	the sub-matrix of \mathbf{K} , with row indices S and column indices R
\mathbf{K}_{sR}	the sub row vector of \mathbf{K} , with row index s and column indices R
$\Sigma^{(S)}$	the matrix Σ when a feature set S is selected.
\mathbf{I}_p	an identity matrix of size $p \times p$
\mathbf{x}	a column vector
\mathbf{x}^\top	the transposition of vector \mathbf{x}
$\mathbf{1}$	a vector whose elements are all ones
$\ \mathbf{x}\ ^2$	the square of the norm of \mathbf{x} , i.e., $\mathbf{x}^\top \mathbf{x}$
λ	a scalar. regularization parameter.
$ D $	the cardinality of set D .
D	the full index set in \mathbf{Z} . $ D = d$.
F	the index set in \mathbf{Z} corresponding to the features, \mathbf{X} . $ F = f$.
T	the index set in \mathbf{Z} corresponding to the targets, \mathbf{Y} . $ T = t$.
S	the index set of selected features.

The prediction capability of Y given X_S can be measured by the entropy of Y given X_S , which is defined as

$$H(Y|X_S) \stackrel{\text{def}}{=} -\mathbb{E}_{p(Y, X_S)}(\ln p(Y|X_S)), \quad (1)$$

where $\mathbb{E}_p(\cdot)$ is the expectation given the distribution p , and p stands for the underlying data distribution, i.e., the joint distribution $p(Y, X_S)$. The feature selection problem using the mutual information criterion is

$$\arg \min_S H(Y|X_S). \quad (2)$$

Selecting an optimal subset of features is a combinatorial optimization problem, which is an NP-problem. For the effective practice is to take a greedy approach, i.e., sequentially selecting features to achieve a suboptimal solution. Given a selected feature set, S , the one-step goal of feature selection is to select one feature to minimize the entropy. The one-step objective, named as information gain, is to find i to maximize

$$\text{IG}(i; S) \stackrel{\text{def}}{=} H(Y|X_S) - H(Y; X_{S \cup \{i\}}).$$

Then, the greedy procedure of feature selection based on mutual information is shown in Algorithm 1.

Algorithm 1. Feature selection by information gain

- 1: Let $S = \emptyset$;
- 2: **repeat**
- 3: $i = \arg \max_{i \in F} \text{IG}(i; S)$;
- 4: $S \leftarrow S \cup \{i\}$;
- 5: **until** $|S| = m$.

The distribution $p(Y, X_S)$ can be estimated by the empirical distribution, i.e., measuring the proportion of Y and X_S values in the given data. The empirical distribution faces the data sparseness problem. Thus, we discuss the

estimation based on a multivariate Gaussian generative model in the next section.

4.2 Multivariate Gaussian Model

We assume that the joint distribution of $\{X_i\}$ and Y is a multivariate normal (Gaussian) distribution:

$$\mathbf{z} = [X_F Y] \sim \mathcal{N}(\cdot; \boldsymbol{\mu}, \boldsymbol{\Sigma}), \quad (3)$$

where $\boldsymbol{\mu}$ is the mean vector, and $\boldsymbol{\Sigma}$ is the covariance matrix. Let F be the index set of X in \mathbf{z} and T be the index set of Y in \mathbf{z} . The reason is that the Gaussian assumption results in a linear model, which is simple and scalable.

We denote the feature values in the training data by $\tilde{\mathbf{X}}$, where each row represents a sample, and each column represents a feature (a gene). We consider multiple target variables. For training data, we denote the target values as $\tilde{\mathbf{Y}}$, where each row represents target variables of a sample, and each column represents a target variable.

Given the training data, we can estimate the parameters of (3) by

$$\hat{\boldsymbol{\mu}} \stackrel{\text{def}}{=} \frac{1}{n} \mathbf{1}^\top [\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}], \quad \hat{\boldsymbol{\Sigma}} \stackrel{\text{def}}{=} \frac{1}{n} \mathbf{Z}^\top \mathbf{Z}, \quad (4)$$

where n is the number of rows of matrix $\tilde{\mathbf{X}}$, $\mathbf{1}$ is a column vector of size n , whose elements are all ones, and

$$\mathbf{Z} \stackrel{\text{def}}{=} [\mathbf{X}, \mathbf{Y}] \stackrel{\text{def}}{=} [\tilde{\mathbf{X}}, \tilde{\mathbf{Y}}] - \mathbf{1} \hat{\boldsymbol{\mu}}^\top. \quad (5)$$

Though the estimation of (4) is an unbiased estimation of the covariance matrix, such estimation may suffer ill-posed problems. By adding a regularization term, a robust estimation can be obtained:

$$\hat{\boldsymbol{\Sigma}} = \frac{1}{n} (\mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{I}_d). \quad (6)$$

This is first proposed in [30]. Since (4) is a special case of (6), we use (6) as the estimation of $\boldsymbol{\Sigma}$.

After the parameters of the model being estimated, we do not differentiate the parameters and the estimated parameters. For simplicity, we write $\hat{\boldsymbol{\Sigma}}$ by $\boldsymbol{\Sigma}$. Since the computation of the entropy does not involve $\boldsymbol{\mu}$, we let $\boldsymbol{\mu} = \mathbf{0}$ without loss of generality. Let \mathbf{z} be a d -dimensional vector, following the multivariate Gaussian distribution $\mathcal{N}(\mathbf{z}; \mathbf{0}, \boldsymbol{\Sigma})$. There are some properties of multivariate Gaussian distribution.

Property 1. Let \mathbf{z}_S and \mathbf{z}_T be two subvectors of \mathbf{z} , where S and T are the index sets. We denote the subvector of $\boldsymbol{\mu}$ corresponding to an index set S by $\boldsymbol{\mu}_S$ and the submatrix of $\boldsymbol{\Sigma}$ corresponding to index sets S and T by $\boldsymbol{\Sigma}_{ST}$. We have

$$\Pr(\mathbf{z}_T | \mathbf{z}_S) \stackrel{\text{def}}{=} \mathcal{N}(\mathbf{z}_T; \boldsymbol{\mu}_{T|S}, \boldsymbol{\Sigma}_{T|S}),$$

where

$$\boldsymbol{\mu}_{T|S} \stackrel{\text{def}}{=} \boldsymbol{\mu}_T + \boldsymbol{\Sigma}_{TS} \boldsymbol{\Sigma}_{SS}^{-1} (\mathbf{z}_S - \boldsymbol{\mu}_S), \quad (7)$$

$$\boldsymbol{\Sigma}_{T|S} \stackrel{\text{def}}{=} \boldsymbol{\Sigma}_{TT} - \boldsymbol{\Sigma}_{TS} \boldsymbol{\Sigma}_{SS}^{-1} \boldsymbol{\Sigma}_{ST}. \quad (8)$$

This is the property of conditional distribution [26].

Then, the property of the incremental updates is given as follows:

Property 2. Let D be the full index set of \mathbf{z} , $S \subset F$, $i \in F - S$.

We have

$$\Sigma_{T|S \cup \{i\}} = \Sigma_{TT}^{(S)} - \frac{1}{\Sigma_{ii}^{(S)}} \Sigma_{Ti}^{(S)} \Sigma_{iT}^{(S)}, \quad (9)$$

where

$$\Sigma^{(S)} \stackrel{\text{def}}{=} \Sigma - \Sigma_{DS} \Sigma_{SS}^{-1} \Sigma_{SD}. \quad (10)$$

Proof. Let $T' = T \cup \{i\}$. By Property 1, $\Pr(\mathbf{z}_{T'}|\mathbf{z}_S)$ follows a Gaussian distribution with covariance $\Sigma_{T'|S}$ (see the definition in (8)). By the definition of $\Sigma^{(S)}$, $\Sigma_{T'|S}$ is the submatrix of $\Sigma^{(S)}$, whose column and row indices are T' . Since $\Pr(\mathbf{z}_T|\mathbf{z}_{S \cup \{i\}}) = \Pr(\mathbf{z}_T|\mathbf{z}_{\{i\}}, \mathbf{z}_S)$, applying Property 1 again, we obtain

$$\Sigma_{T|S \cup \{i\}} = \Sigma_{TT}^{(S)} - \Sigma_{Ti}^{(S)} \left[\Sigma_{ii}^{(S)} \right]^{-1} \Sigma_{iT}^{(S)}.$$

As $\Sigma_{ii}^{(S)}$ is a scalar, we have (9). \square

The differential entropy of the multivariate normal distribution can be computed by the following property.

Property 3.

$$\begin{aligned} H(\mathbf{z}) &= - \int \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \Sigma) \ln \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}, \Sigma) d\mathbf{z} \\ &= \frac{1}{2} \ln |\Sigma| + \frac{d}{2} \ln(2\pi e). \end{aligned}$$

This can be found in [26].

Property 4.

$$H(Y|X_S) = \frac{1}{2} \ln |\Sigma_{T|S}| + \frac{d}{2} \ln(2\pi e), \quad (11)$$

where T is its set of indices for Y in \mathbf{z} .

By Properties 1 and 3, we have the property of the conditional multivariate normal distribution.

4.3 Feature Selection Algorithms

Now, we propose two sets of feature selection algorithms based on the multivariate Gaussian model and entropy measure.

4.3.1 D -Optimality Feature Selection

In the multivariate Gaussian model, the problem of feature selection (2) becomes

$$\arg \min_S \ln |\Sigma_{T|S}|. \quad (12)$$

As the determinant of the covariance matrix is known as *generalized variance*. This criterion is to minimize the generalized variance of the joint distribution of targets. We name the feature selection based on the determinant criterion (12) as the *D-optimality Feature Selection* after the determinant. We borrow the name of *D-optimality* from experimental designs [9].

As solving (12) is still an NP-problem, we use the greedy approach as in Algorithm 1. Let $\Sigma^{(S)} = \mathbf{K} + \lambda \mathbf{I}_d$. By (9), we have

$$\begin{aligned} \ln |\Sigma_{T|S \cup \{i\}}| &= \ln \left| \Sigma_{TT}^{(S)} - \frac{1}{\Sigma_{ii}^{(S)}} \left(\Sigma_{Ti}^{(S)} \Sigma_{iT}^{(S)} \right) \right| \\ &= \ln \left| \mathbf{K}_{TT} + \lambda \mathbf{I}_t - \frac{1}{\mathbf{K}_{ii} + \lambda} (\mathbf{K}_{Ti} \mathbf{K}_{iT}) \right| \\ &= \ln |\mathbf{K}_{TT} + \lambda \mathbf{I}_t| + \ln \left(1 - \frac{\mathbf{K}_{iT} (\mathbf{K}_{TT} + \lambda \mathbf{I}_t)^{-1} \mathbf{K}_{Ti}}{\mathbf{K}_{ii} + \lambda} \right), \end{aligned} \quad (13)$$

where $t = |T|$. Therefore,

$$\arg \min_i \ln |\Sigma_{T|S \cup \{i\}}| = \arg \max_i \frac{\mathbf{K}_{iT} (\mathbf{K}_{TT} + \lambda \mathbf{I}_t)^{-1} \mathbf{K}_{Ti}}{\mathbf{K}_{ii} + \lambda}. \quad (14)$$

We can compute $\Sigma^{(S \cup \{i\})}$ from $\Sigma^{(S)}$ by (10):

$$\begin{aligned} \Sigma^{(S \cup \{i\})} &= \Sigma^{(S)} - \frac{1}{\Sigma_{ii}^{(S)}} \left(\Sigma_{Di}^{(S)} \Sigma_{iD}^{(S)} \right) \\ &= \mathbf{K} + \lambda \mathbf{I}_d - \frac{1}{\mathbf{K}_{ii} + \lambda} \\ &\quad (\mathbf{K}_{Di} \mathbf{K}_{iD} + \lambda \delta_i \mathbf{K}_{iD} + \lambda \mathbf{K}_{Di} \delta_i^\top + \lambda^2 \delta_i \delta_i^\top), \end{aligned} \quad (15)$$

where δ_i is a column vector whose elements are zeros except that the i th element is one. Since we shall not select a feature twice, we shall no longer be concerned with the values in the i th row or column in \mathbf{K} . Therefore, we can update \mathbf{K} by $\mathbf{K} - \frac{1}{\mathbf{K}_{ii} + \lambda} (\mathbf{K}_{Di} \mathbf{K}_{iD})$. By sequentially updating \mathbf{K} , we have Algorithm 2. Note that since we only compare the values for features, we can drop the scale factor $\frac{1}{n}$ in (6) for simplicity. The complexity of Algorithm 2 is $O(m(d^2 + dt^2))$, where $d = |D|$, and $t = |T|$.

Since the complexity of the algorithm contains md^2 , the algorithm is very inefficient when d is large. Especially, the memory complexity is $O(d^2)$. When the sample size n is much smaller than d , we can take advantage of it to speed up the algorithm. Assume that \mathbf{K} has the form of $\mathbf{Z}^\top \Phi \mathbf{Z}$. Note that Φ is symmetric since \mathbf{K} is symmetric. Initially, $\Phi = \mathbf{I}_n$ in step 2 of Algorithm 2.

Algorithm 2. D -Optimality feature selection I

- 1: $S = \emptyset$;
- 2: $\mathbf{K} = \mathbf{Z}^\top \mathbf{Z}$;
- 3: **repeat**
- 4: Let $\mathbf{U} = (\mathbf{K}_{TT} + \lambda \mathbf{I}_t)^{-1}$;
- 5: $i = \arg \max_{i \in F-S} \frac{\mathbf{K}_{iT} \mathbf{U} \mathbf{K}_{Ti}}{\mathbf{K}_{ii} + \lambda}$;
- 6: $\mathbf{K} \leftarrow \mathbf{K} - \frac{1}{\mathbf{K}_{ii} + \lambda} (\mathbf{K}_{Di} \mathbf{K}_{iD})$;
- 7: $S \leftarrow S \cup \{i\}$;
- 8: **until** $|S| = m$.

Let us denote the i th column vector of \mathbf{X} as \mathbf{x}_i . The update of \mathbf{K} can be written as

$$\mathbf{K} - \frac{1}{\mathbf{K}_{ii} + \lambda} (\mathbf{K}_{Di} \mathbf{K}_{iD}) = \mathbf{Z}^\top \Phi \mathbf{Z} - \frac{1}{\mathbf{x}_i^\top \Phi \mathbf{x}_i + \lambda} (\mathbf{Z}^\top \Phi \mathbf{x}_i \mathbf{x}_i^\top \Phi \mathbf{Z}). \quad (16)$$

Therefore, we derive the update for Φ in Algorithm 3. The complexity of the algorithm is $O(m(dn^2 + nt^2 + t^3))$.

Algorithm 3. *D*-Optimality feature selection II

- 1: $S = \emptyset$;
- 2: $\Phi = \mathbf{I}_n$;
- 3: **repeat**
- 4: Let $\Omega = \Phi \mathbf{Y} (\mathbf{Y}^\top \Phi \mathbf{Y} + \lambda \mathbf{I}_t)^{-1} \mathbf{Y}^\top \Phi$;
- 5: $i = \arg \max_{i \in F-S} \frac{\mathbf{x}_i^\top \Omega \mathbf{x}_i}{\mathbf{x}_i^\top \Phi \mathbf{x}_i + \lambda}$;
- 6: $\Phi \leftarrow \Phi - \frac{1}{\mathbf{x}_i^\top \Phi \mathbf{x}_i + \lambda} (\Phi \mathbf{x}_i \mathbf{x}_i^\top \Phi)$;
- 7: $S \leftarrow S \cup \{i\}$;
- 8: **until** $|S| = m$.

N.B. \mathbf{x}_i is the i th column of centered feature matrix \mathbf{X} .

When $t < n$, we can reduce the complexity by sequentially computing $\Phi \mathbf{X} \stackrel{\text{def}}{=} \mathbf{P}$. We have Algorithm 4, whose complexity is $O(mdnt)$. Note that *Phi* and *P* in Algorithm 3 and Algorithm 4 are used to save the intermediate results in the matrix computation to reduce the computation complexity.

Algorithm 4. *D*-Optimality feature selection III

- 1: $S = \emptyset$;
- 2: $\Phi = \mathbf{I}_n$;
- 3: $\mathbf{P} = \mathbf{X}$;
- 4: **repeat**
- 5: Let $\mathbf{R} = \mathbf{Y} ((\mathbf{Y}^\top \Phi \mathbf{Y} + \lambda \mathbf{I}_t)^{-1} \mathbf{Y}^\top \mathbf{P})$;
- 6: $i = \arg \max_{i \in F-S} \frac{\mathbf{r}_i^\top \mathbf{p}_i}{\mathbf{x}_i^\top \mathbf{p}_i + \lambda}$;
- 7: $\Phi \leftarrow \Phi - \frac{1}{\mathbf{x}_i^\top \mathbf{p}_i + \lambda} (\mathbf{p}_i \mathbf{p}_i^\top)$;
- 8: $\mathbf{P} \leftarrow \mathbf{P} - \frac{1}{\mathbf{x}_i^\top \mathbf{p}_i + \lambda} (\mathbf{p}_i (\mathbf{p}_i^\top \mathbf{X}))$;
- 9: $S \leftarrow S \cup \{i\}$;
- 10: **until** $|S| = m$.

N.B. \mathbf{x}_i is the i th column of centered feature matrix \mathbf{X} . \mathbf{p}_i is the i th column of \mathbf{P} , and \mathbf{r}_i is the i th column of \mathbf{R} .

4.3.2 A-Optimality Feature Selection

Because of the complexity in computing determinants and the nonconvexity of log-determinants, we can replace the log-determinant of the covariance matrix with the trace of the covariance matrix, which is the upper bound of the log-determinant of the covariance matrix.

Lemma 1. *If \mathbf{X} is a $p \times p$ positive definite matrix, it holds that $\ln |\mathbf{X}| \leq \text{tr}(\mathbf{X}) - p$. The equality holds when \mathbf{X} is an orthonormal matrix.*

Proof. Let $\{\lambda_1, \dots, \lambda_p\}$ be the eigenvalues of \mathbf{X} . We have $\ln |\mathbf{X}| = \sum_i \ln \lambda_i$ and $\text{tr}(\mathbf{X}) = \sum_i \lambda_i$. Since $\ln \lambda_i \leq \lambda_i - 1$, we have the inequality. The equality holds when $\lambda_i = 1$. Therefore, when \mathbf{X} is an orthonormal matrix (especially $\mathbf{X} = \mathbf{I}_p$), the equality holds. \square

As $\ln |\Sigma_{T|S}| \leq \text{tr}(\Sigma_{T|S}) - t$, the problem of feature selection (12) can be approximated by

$$\arg \min_s \text{tr}(\Sigma_{T|S}) = \arg \max_s \text{tr}(\Sigma_{TS} \Sigma_{SS}^{-1} \Sigma_{ST}). \quad (17)$$

Since the trace of the covariance divided by the number of variables is the *average* covariance, (17) is called *A-optimality feature selection*. We also borrow the name of *A-optimality* from experimental designs, which is an alternative of the *D-optimality* criterion [9].

To sequentially solve (17), we have Algorithm 5. The algorithm is similar to the sequential algorithm in [35],

TABLE 2
The Complexity of Algorithms

Algorithm	Time	Space
<i>D</i> -opt I	$O(m(d^2 + dt^2))$	$O(d^2)$
<i>D</i> -opt II	$O(m(dn^2 + nt^2 + t^3))$	$O(n^2 + d)$
<i>D</i> -opt III	$O(mdnt)$	$O(n^2 + nd)$
<i>A</i> -opt I	$O(m(d^2 + dt^2))$	$O(d^2)$
<i>A</i> -opt II	$O(mdn^2)$	$O(n^2 + d)$
<i>A</i> -opt III	$O(mdn + dnt)$	$O(nd)$

The space complexity measures the extra required space besides data.

which is to solve transductive active learning problems. The complexity of Algorithm 5 is $O(md^2)$.

Algorithm 5. *A*-Optimality feature selection I

- 1: $S = \emptyset$;
- 2: $\mathbf{K} = \mathbf{Z}^\top \mathbf{Z} + \lambda \mathbf{I}_d$;
- 3: **repeat**
- 4: $i = \arg \max_{i \in F-S} \frac{\mathbf{K}_{ii} \mathbf{K}_{ii}}{\mathbf{K}_{ii} + \lambda}$;
- 5: $\mathbf{K} \leftarrow \mathbf{K} - \frac{1}{\mathbf{K}_{ii} + \lambda} (\Sigma_{D_i} \mathbf{K}_{iD})$;
- 6: $S \leftarrow S \cup \{i\}$;
- 7: **until** $|S| = m$.

When $n \ll d$, we can use a similar method as in Algorithm 3 to speed up Algorithm 5. We can obtain Algorithm 5 by letting $\mathbf{U} = \mathbf{I}_t$ in Algorithm 2. Then, we can use

$$\Omega = \Phi \mathbf{Y} \mathbf{Y}^\top \Phi$$

in step 4 of Algorithm 3 to obtain Algorithm 6. The complexity of Algorithm 6 is $O(mdn^2)$.

Algorithm 6. *A*-Optimality feature selection II

- 1: $S = \emptyset$;
- 2: $\Phi = \mathbf{I}_n$;
- 3: **repeat**
- 4: Let $\Omega = \Phi \mathbf{Y} \mathbf{Y}^\top \Phi$;
- 5: $i = \arg \max_{i \in F-S} \frac{\mathbf{x}_i^\top \Omega \mathbf{x}_i}{\mathbf{x}_i^\top \Phi \mathbf{x}_i + \lambda}$;
- 6: $\Phi \leftarrow \Phi - \frac{1}{\mathbf{x}_i^\top \Phi \mathbf{x}_i + \lambda} (\Phi \mathbf{x}_i \mathbf{x}_i^\top \Phi)$;
- 7: $S \leftarrow S \cup \{i\}$;
- 8: **until** $|S| = m$.

N.B. \mathbf{x}_i is the i th column of centered feature matrix \mathbf{X} .

We can also sequentially compute $\Phi \mathbf{X} \stackrel{\text{def}}{=} \mathbf{P}$ as shown in Algorithm 7, whose time complexity is $O(mdn + dnt)$.

Algorithm 7. *A*-Optimality feature selection III

- 1: $S = \emptyset$;
- 2: $\mathbf{P} = \mathbf{X}$;
- 3: $\mathbf{R} = \mathbf{Y} (\mathbf{Y}^\top \mathbf{P})$;
- 4: **repeat**
- 5: $i = \arg \max_{i \in F-S} \frac{\mathbf{r}_i^\top \mathbf{p}_i}{\mathbf{x}_i^\top \mathbf{p}_i + \lambda}$;
- 6: $\mathbf{R} \leftarrow \mathbf{R} - \frac{1}{\mathbf{x}_i^\top \mathbf{p}_i + \lambda} (\mathbf{r}_i (\mathbf{p}_i^\top \mathbf{X}))$;
- 7: $\mathbf{P} \leftarrow \mathbf{P} - \frac{1}{\mathbf{x}_i^\top \mathbf{p}_i + \lambda} (\mathbf{p}_i (\mathbf{p}_i^\top \mathbf{X}))$;
- 8: $S \leftarrow S \cup \{i\}$;
- 9: **until** $|S| = m$.

N.B. \mathbf{x}_i is the i th column of centered feature matrix \mathbf{X} . \mathbf{p}_i is the i th column of \mathbf{P} , and \mathbf{r}_i is the i th column of \mathbf{R} .

Table 2 shows the summary of the complexity of the above algorithms. In the gene expression data, which

contain a large number of genes but a small number of samples, the D -opt III and A -opt III are the good choice for computational efficiency.

5 METHODS USED FOR COMPARISON

In this section, we describe several feature selection methods used in our experimental comparisons.

5.1 Rankgene

We use the following feature selection methods provided in the program Rankgene [31]: *information gain*, *twoing rule*, and *sum minority*. These methods have been widely used either in machine learning (information gain) or in statistical learning theory (twoing rule and sum minority). All these methods measure the effectiveness of a feature by evaluating the strength of class prediction when the prediction is made by splitting it into two regions, the high region and the low region, by considering all possible split points.

5.2 Max-Relevance

The Max-Relevance method selects a set of genes with the highest relevance to the target class [25]. Given g_i , which represents the gene i , and the class label c , their mutual information is defined in terms of their frequencies of appearances $p(g_i)$, $p(c)$, and $p(g_i, c)$ as follows:

$$I(g_i, c) = \iint p(g_i, c) \ln \frac{p(g_i, c)}{p(g_i)p(c)} dg_i dc. \quad (18)$$

The Max-Relevance method selects the top m genes in the descent order of $I(g_i, c)$, i.e., the best m individual features correlated to the class labels.

5.3 mRMR

Although we can choose the top individual genes using the Max-Relevance algorithm, it has been recognized that “the m best features are not the best m features” since the correlations among those top features may also be high [7]. In order to remove the redundancy among features, an mRMR framework is proposed in [25]. In mRMR, the mutual information between each pair of genes is taken into consideration. Suppose set G represents the set of genes and we already have S_{m-1} , the feature set with $m-1$ genes; then, the task is to select the m th feature from the set $\{G - S_{m-1}\}$. In the following formula, we see that minimizing the redundancy and maximizing the relevance can be achieved concordantly [25] (the methods proposed in [36] share a similar idea with mRMR):

$$\max_{g_j \in G - S_{m-1}} \left[I(g_j; c) - \frac{1}{m-1} \sum_{g_i \in S_{m-1}} I(g_j; g_i) \right]. \quad (19)$$

5.4 ReliefF

ReliefF is a simple yet efficient procedure to estimate the quality of attributes in problems with strong dependencies between attributes [22]. In practice, ReliefF is usually applied as a feature subset selection method.

The key idea of the ReliefF is to estimate the quality of genes according to how well their values distinguish between instances that are near to each other. Given a randomly selected instance Ins_m from class L , ReliefF searches for K of

TABLE 3
The Data Set Description

Dataset	# Samples	# Genes	# Classes
ALL	248	12558	6
GCM	198	16063	14
HBC	22	3226	3
Lymphoma	62	4026	3
MLL	72	12582	3
NCI60	60	1123	9
SRBCT	83	2308	4

its nearest neighbors from the same class, called nearest hits H , and also K nearest neighbors from each of the different classes, called nearest misses M . It then updates the quality estimation W_i for gene i based on their values for Ins_m , H , and M . If instance Ins_m and those in H have different values on gene i , then the quality estimation W_i is decreased. On the other hand, if instance Ins_m and those in M have different values on gene i , then W_i is increased. The whole process is repeated n times, which is set by users. The equation below can be used to update W_i :

$$W_i = W_i - \frac{\sum_{k=1}^K D_H}{n \cdot K} + \sum_{c=1}^{C-1} P_c \cdot \frac{\sum_{k=1}^K D_{M_c}}{n \cdot K}, \quad (20)$$

where n_c is the number of instances in class c , D_H (or D_{M_c}) is the sum of distance between the selected instance and each H (or M_c), and P_c is the prior probability of class c . Detailed discussions on ReliefF can be found in [22].

5.5 D -opt and A -opt Methods

Consider the large number of features (genes) and the relative small number of samples, we use Algorithm 4 for the D -optimality feature selection (denoted by D -opt) and Algorithm 7 for the A -optimality feature selection (denoted by A -opt). The mean of each data set is removed as shown in (5). Note that the standardization is a way of increasing the degree of normality for the gene expression data [34]. The regularization parameter, λ , is set to 0.5 in our experiments.

6 EXPERIMENTS

We conduct three sets of experiments using seven data sets as described in Section 6.1. In the first set of experiments, we compare the classification accuracy of data with gene selection and without gene selection using Support Vector Machine (SVM) classifiers implemented in the LIBSVM package [4]. The second set of experiments provides a comprehensive study on the performance of different gene selection methods under different conditions. In the third set of experiments, we discuss the number of selected genes.

6.1 Data Set Description

The data sets and their characteristics are summarized in Table 3.

The ALL data set [33] is a data set that covers six subtypes of acute lymphoblastic leukemia: BCR (15), E2A (27), Hyperdip (64), MLL (20), T (43), and TEL (79). Here, the numbers in the parentheses are the numbers of samples. The data set is available at [2]. The GCM data set [27]

TABLE 4
Comparative Accuracy of Different Selection Methods on Seven Data Sets (Gene Number = 30)

	SRBCT	NCI60	Lymphoma	GCM	HBC	ALL	MLL
No Gene Selection	85.22%	63.33%	95.16%	51.52%	77.27%	91.94%	97.22%
mRMR	81.82%	53.33%	100.0%	N/A	95.45%	N/A	N/A
Maxrel	84.09%	51.67%	100.0%	60.61%	72.73%	89.11%	77.78%
ReliefF	89.77%	58.33%	100.0%	55.25%	95.45%	96.37%	94.44%
Information Gain	89.77%	61.67%	98.39%	46.97%	100.0%	97.58%	98.67%
Sum Minority	78.41%	65.00%	98.39%	55.05%	95.45%	93.95%	90.28%
Twoing Rule	84.09%	61.67%	98.39%	45.96%	90.91%	96.77%	97.22%
A-opt (Alg. 7)	94.32%	88.33%	100.0%	75.25%	100.0%	99.19%	100.0%
D-opt (Alg. 4)	90.91%	80.00%	100.0%	73.23%	100.0%	100.0%	100.0%

Because of limitation in memory, mRMR cannot run on GCM, ALL, and MLL.

consists of 198 human tumor samples of 15 types. The HBC data set consists of 22 hereditary breast cancer samples and was first studied in [12]. The data set has three classes and can be downloaded at [11]. The Lymphoma data set is a data set of the three most prevalent adult lymphoid malignancies and available at [20], and it was first studied in [1]. The MLL-leukemia data set consists of three classes

and can be downloaded at [21]. The NCI60 data set was first studied in [28]. cDNA microarrays were used to examine the variation in gene expression among the 60 cell lines from the National Center Institute’s anticancer drug screen. The data set spans nine classes and can be downloaded at [23]. SRBCT [14] is the data set of small round blue cell tumors of childhood and can be downloaded at [29].

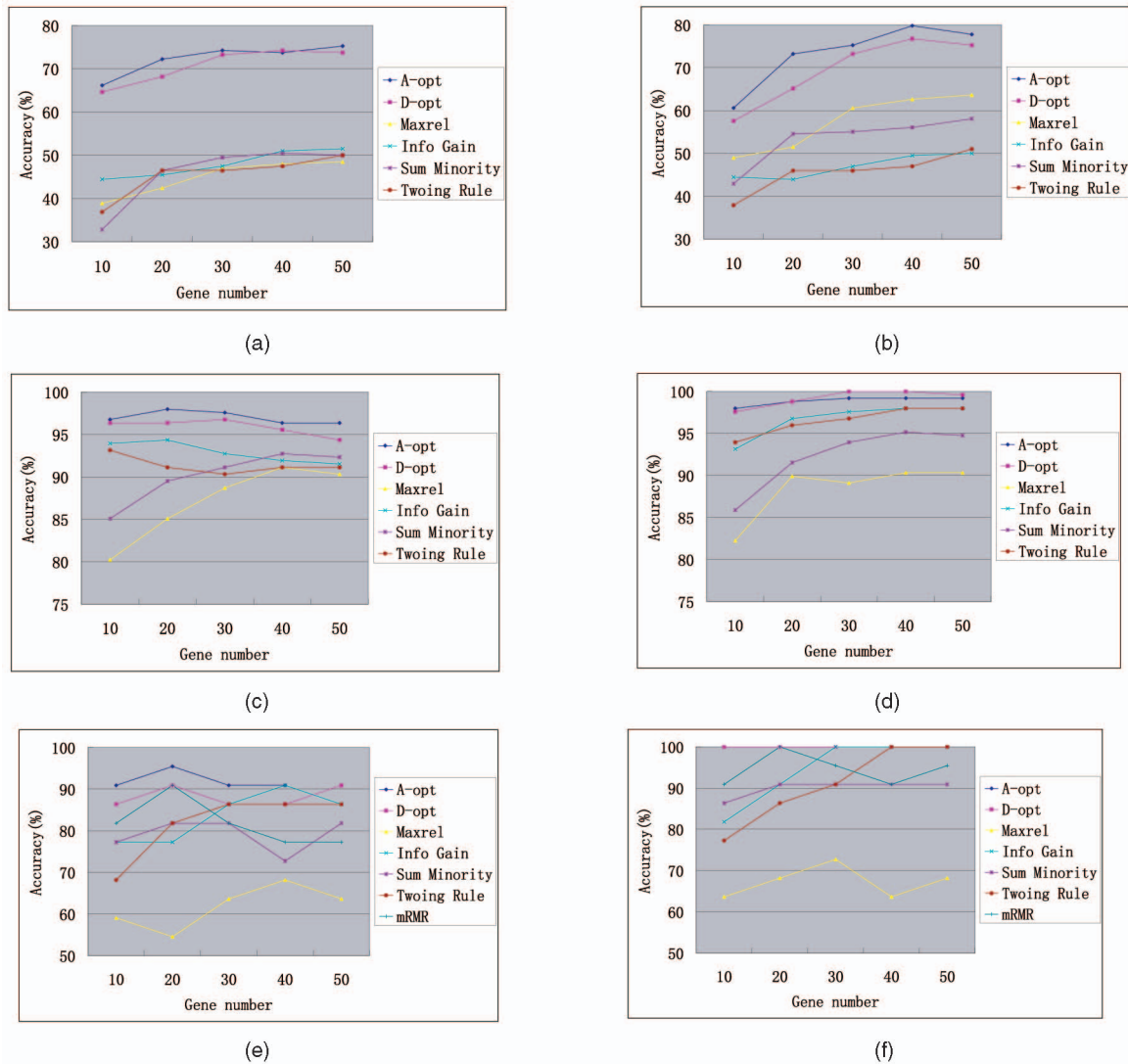


Fig. 1. Comparison of various gene selection methods (I). (a) Results of Naive Bayes: GCM data set. (b) Results of SVM: GCM data set. (c) Results of Naive Bayes: ALL data set. (d) Results of SVM: ALL data set. (e) Results of Naive Bayes: HBC data set. (f) Results of SVM: HBC data set.

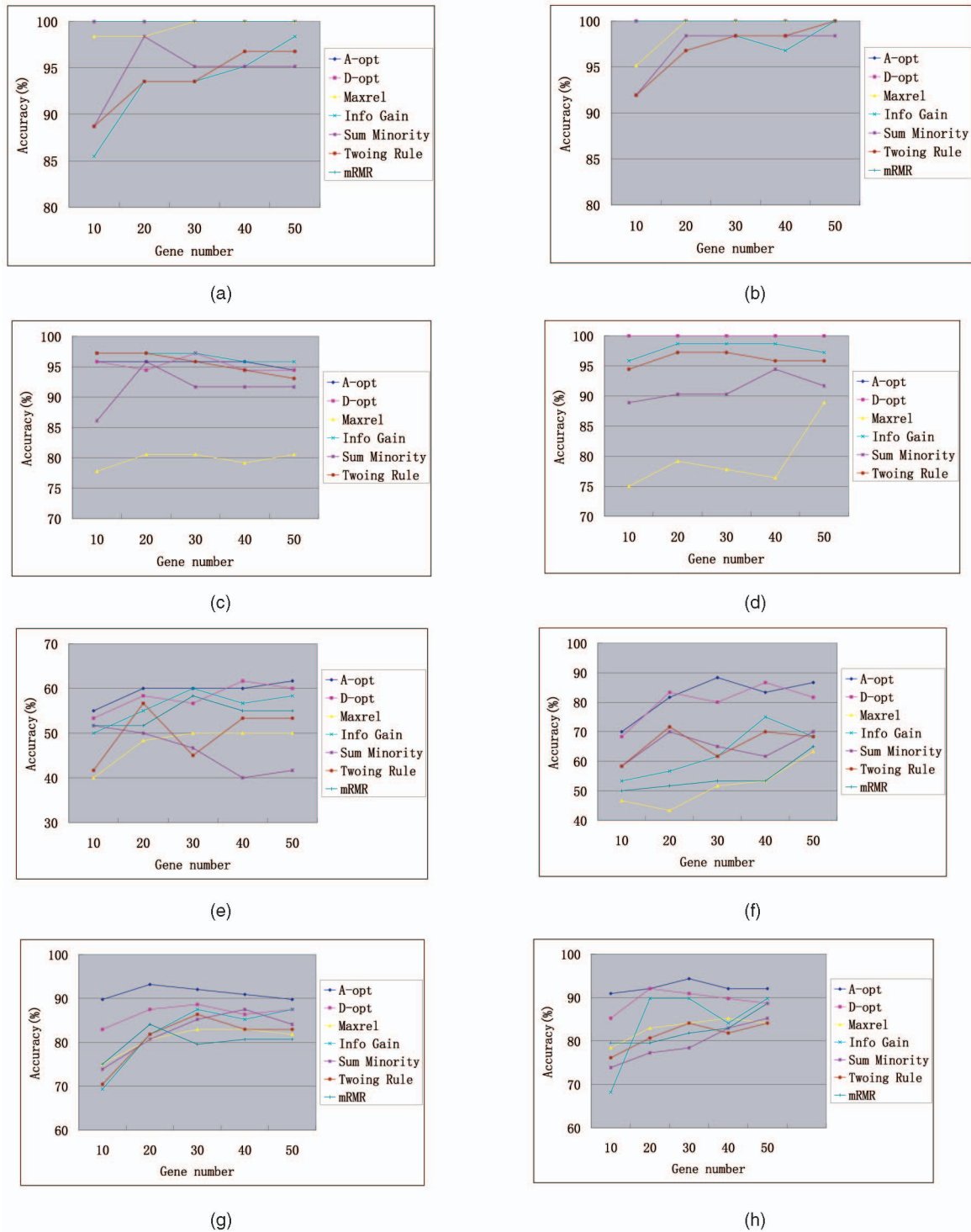


Fig. 2. Comparison of various gene selection methods (II). (a) Results of Naive Bayes: Lymphoma data set. (b) Results of SVM: Lymphoma data set. (c) Results of Naive Bayes: MLL data set. (d) Results of SVM: MLL data set. (e) Results of Naive Bayes: NCI60 data set. (f) Results of SVM: NCI60 data set. (g) Results of Naive Bayes: SRBCT data set. (h) Results of SVM: SRBCT data set.

6.2 Effectiveness of Gene Selection

Table 4 presents the accuracy values of applying SVM on the top 30 genes selected by different methods and also on all the genes without selection. The accuracy values are obtained via 10-fold cross validation. The table shows that gene selection improves classification performance; at least the accuracy of SVM on genes selected by both the *D*-opt and *A*-opt methods outperform that without feature

selection. We will discuss the number of selected genes in Section 6.4.

6.3 Performance of Different Gene Selection Methods

In this section, we present a comparative study of various gene selection methods using SVM and Naive Bayes algorithms on the seven data sets. Both SVM and Naive

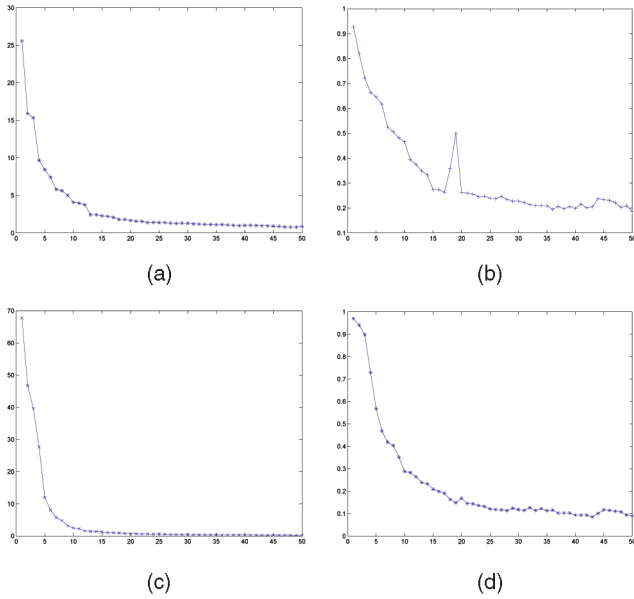


Fig. 3. Variance reduction on data sets (I). (a) A -opt: GCM data set. (b) D -opt: GCM data set. (c) A -opt: ALL data set. (d) D -opt: ALL data set.

Bayes have been widely used in previous studies(e.g., [18] and [25]). Figs. 1 and 2 show the classification accuracy results as a function of the number of selected genes on the seven data sets, respectively. From the comparative study, we observe the following:

- Gene selection by experimental design (D -opt and A -opt) outperforms other gene selection methods such as information gain, etc. It largely owes this to the generality of the multivariate Gaussian generative model. In addition, our methods estimate the information gain based on models, instead of on the data itself. This overcomes the limitations of data sparseness and provides more robust and accurate estimations.
- The results of the A -opt method are similar to those of the D -opt method. Besides the simplicity of A -opt, the A -opt method outperforms the D -opt method in most cases. There are some discussion of comparing A -optimality and D -optimality in the literature experimental designs [9].
- Gene selection by D -opt and A -opt implicitly selects the features with the minimum redundancy. In step 6 of Algorithm 2 and step 5 of Algorithm 5, the covariance matrices are updated, which removes the second-order redundancy. We can find similar actions in other algorithms as well.

6.4 Number of Selected Genes

From the above experiment, it can be observed that when the number of selected genes is greater than 30, the variation of the performance is small. In step 6 of Algorithm 4, we select genes to reduce the generalized variance. In step 5 of Algorithm 7, we select genes to reduce the total variance. Figs. 3, 4, and 5 show the variance reduction as the function of the number of genes on the seven data sets, respectively. The number of selected genes is varied from 1 to 50, and the results show the change of classification accuracy.

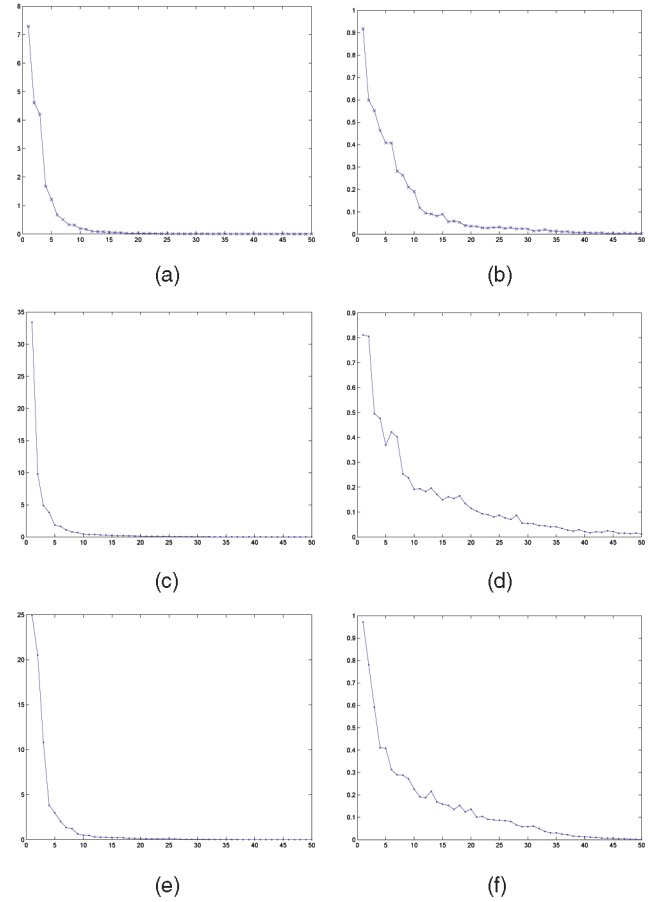


Fig. 4. Variance reduction on data sets (II). (a) A -opt: HBC data set. (b) D -opt: HBC data set. (c) A -opt: Lymphoma data set. (d) D -opt: Lymphoma data set. (e) A -opt: MLL data set. (f) D -opt: MLL data set.

The experiment results demonstrate that only a small number of genes are needed for classification purposes. In our experiments, we observe that when the number of selected genes is greater than 30, the variation of the classification performance is small. We find that the cumulative reduction in generalized variance or total variance converges after 30 steps.

6.5 Other Discussion

This set of experiments aims to study the choice of the regularization parameter λ in our proposed A -opt and D -opt methods. We set the number of selected genes to be 30 and change λ from 0.1 to 0.9. Fig. 6 shows that the accuracy is not sensitive to the regularization parameter. Note that on the LYM and HBC data sets, the accuracies of both methods are 100 percent under different regularization parameters. In our experiments, we choose 0.5 as λ .

7 DISCUSSIONS

Though we are studying the feature selection problem in this paper, the idea largely owes to that of experimental designs.

In the statistics literature, the *experimental designs* can be backtracked to the ideas presented in [15]. The goal of experimental designs is usually to extract the maximum amount of information from as few observations as possible. For experimental designs, several criteria can be

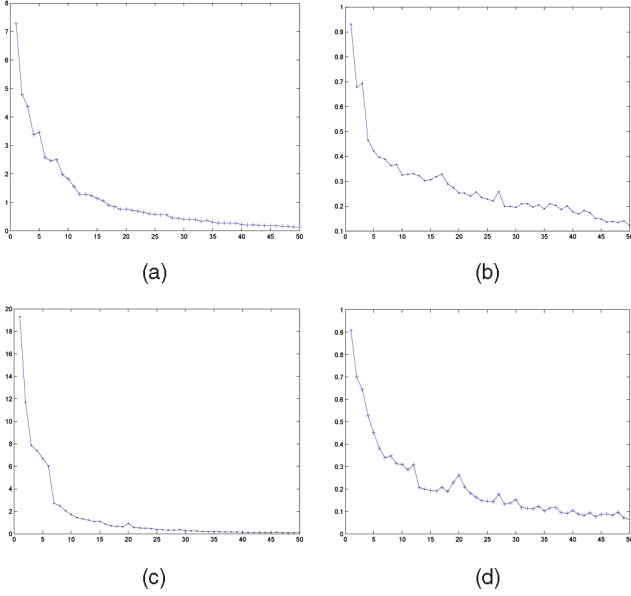


Fig. 5. Variance reduction on data sets (III). (a) A -opt: NCI60 data set. (b) D -opt: NCI60 data set. (c) A -opt: SRBCT data set. (d) D -opt: SRBCT data set.

used, such as D -optimality and A -optimality [9]. They all concern about reducing the uncertainties of estimated parameters. The criterion of the D -optimality is minimizing the generalized variance of the joint distribution of parameters, i.e., the *determinant* of the multivariate variance, which gives its name. The criterion of the A -optimality is minimizing the *average* variance of all parameters.

As concentrating on the predictive variance of a target set of data, Yu et al. [35] propose *transductive experimental designs* for least squares linear (or kernel) regression. The idea is to add samples to the training set in order to improve the numerical stability of predictions on the target test data, measured by the inversion of the Fisher information matrix. It has been shown that the predictive stability only depends on the *locations* of the selected training data, while it does not depend on their *label values*, which leads to a very simple active learning approach [35].

Though there is a big difference between experimental designs and feature selection at the first glance, we find a *duality property* between them.

Let us consider the problem of predicting target Y given a row of feature random vectors X . We assume that the model is a linear model:

$$Y = X^T \mathbf{w} + \epsilon, \quad (21)$$

where \mathbf{w} is the weight vector, and ϵ is the error. The reason for using linear models is because linear models are simple and scalable.

Given the training data \mathbf{y} and \mathbf{X} , where \mathbf{y} is the column target vector and \mathbf{X} is the feature matrix, each row of \mathbf{X} is a feature vector. We can write (21) in matrix format as

$$\mathbf{y} = \mathbf{X}\mathbf{w} + \epsilon,$$

where ϵ is the error vector.

We further assume that the loss function is a square loss; therefore, we want to minimize $\frac{1}{2} \epsilon^T \epsilon$. Meanwhile, we prefer

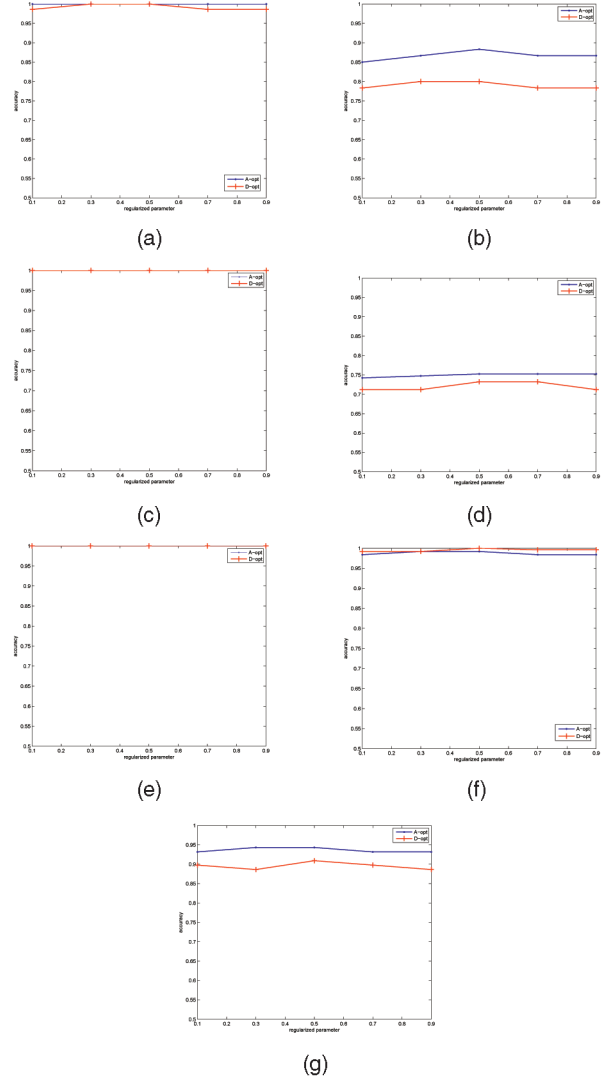


Fig. 6. Different regularization parameters on the seven data sets. (a) MLL data set. (b) NCI60 data set. (c) LYM data set. (d) GCM data set. (e) HBC data set. (f) ALL data set. (g) SRBCT data set.

a robust estimation of \mathbf{w} , i.e., a regularization term, $\frac{\lambda}{2} \mathbf{w}^T \mathbf{w}$. Combining them, the estimation problem becomes

$$\arg \min_{\mathbf{w}} \frac{1}{2} (\mathbf{y} - \mathbf{X}\mathbf{w})^T (\mathbf{y} - \mathbf{X}\mathbf{w}) + \frac{\lambda}{2} \mathbf{w}^T \mathbf{w}. \quad (22)$$

Problem (22) can be explicitly solved as

$$\hat{\mathbf{w}} = (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}. \quad (23)$$

This is also known as ridge regression.

Given a feature vector \mathbf{x} , the estimation of y is

$$\hat{y} = \mathbf{x}^T (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{X}^T \mathbf{y}. \quad (24)$$

On the other hand, we can estimate y by the multivariate Gaussian model. For simplicity, we assume that $\mu = \mathbf{0}$. By (7), we know that

$$\begin{aligned} \hat{y} &= \boldsymbol{\mu}_{T|F} = \boldsymbol{\Sigma}_{TF} (\boldsymbol{\Sigma}_{FF})^{-1} \mathbf{x} \\ &= \mathbf{y}^T \mathbf{X} (\mathbf{X}^T \mathbf{X} + \lambda \mathbf{I})^{-1} \mathbf{x}, \end{aligned}$$

which is equal to (24) as long as we have the same λ .

This shows the duality between the target label and the feature. This property motivates us to treat the feature selection as a dual problem of selecting data samples to label in active learning or experimental designs. Then, we can apply experimental design approaches, more precisely *transductive experimental design* [35], onto the feature selection problem.

8 CONCLUSIONS

In this paper, we suggest multivariate Gaussian generative models for feature (gene) selection because multivariate normal (Gaussian) distributions are a maximum-entropy probability distribution. Using the model-based entropy estimation, we avoid the data sparseness problem that commonly happens in the empirical information gain approach.

Using the properties of multivariate normal distributions, we derive the feature selection methods based on the D -optimality criterion and its approximation, the A -optimality criterion.

To efficiently select genes from gene expression data, where the numbers of features are large and the numbers of samples are relatively small, we propose several simple algorithms (a few lines of code). Among them, Algorithm 4 and Algorithm 7 are most suitable for gene expression data. The time complexity of the proposed algorithms is linear to the product of the number of genes and the number of samples for each iteration of selection.

The experiments on seven gene data sets and the comparison with other five approaches show the accuracy and efficiency of our approach.

ACKNOWLEDGMENTS

The authors would like to thank NIH/NIGMS S06 GM008205, US NSF IIS-0546280, and DBI-0850203 for partially supporting this work.

REFERENCES

- [1] A.A. Alizadeh, M.B. Eisen, R.E. David, C. Ma, I.S. Lossos, A.R. osenwald, H.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Martu, T. Moore, J. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, G.P. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botsten, P.O. Brown, and L.M. Staudt, "Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling," *Nature*, vol. 403, pp. 503-511, 2000.
- [2] ALL, <http://www.stjuderesearch.org/data/ALL1/>, 2008.
- [3] C. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.
- [4] C.-C. Chang and C.-J. Lin, "LIBSVM: A Library for Support Vector Machines," <http://www.csie.ntu.edu.tw/~cjlin/libsvm>, 2001.
- [5] M. Chee, R. Yang, E. Hubbell, A. Berno, X. Huang, D. Stern, J. Winkler, D. Lockhart, M. Morris, and S. Fodor, "Accessing Genetic Information with High Density DNA Arrays," *Science*, vol. 274, pp. 610-614, 1996.
- [6] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, 1991.
- [7] T. Cover, "The Best Two Independent Measurements Are Not the Two Best," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 4, pp. 116-117, 1974.
- [8] S. Dudoit, J. Fridlyand, and T.P. Speed, "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data," *J. Am. Statistical Assoc.*, vol. 97, no. 457, pp. 77-87, 2002.
- [9] V.V. Fedorov, *Theory of Optimal Experiments*. Academic Press, 1972.
- [10] S. Fodor, J. Read, M. Pirrung, L. Stryer, A. Lu, and D. Solas, "Light-Directed, Spatially Addressable Parallel Chemical Synthesis," *Science*, vol. 251, pp. 767-783, 1991.
- [11] HBC, <http://www.columbia.edu/~xy56/project.htm>, 2007.
- [12] I. Hedenfalk, D. Duggan, Y.C. Radmacher, M. Bittner, M. Simon, R. Meltzer, P. Gusterson, B. Esteller, M. Kallioniemi, B.W. Borg, and A. Trent, "Gene-Expression Profiles in Hereditary Breast Cancer," *New England J. Medicine*, vol. 344, no. 8, pp. 539-548, 2001.
- [13] E.T. Jaynes, "Information Theory and Statistical Mechanics," *Physical Rev.*, vol. 106, no. 4, pp. 620-630, May 1957.
- [14] J. Khan, J. Wei, M. Ringner, L. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, and P. Meltzer, "Classification and Diagnostic Prediction of Cancers Using Expression Profiling and Artificial Neural Networks," *Nature Medicine*, vol. 7, no. 6, pp. 673-679, 2001.
- [15] J. Kiefer, "Optimum Experimental Designs," *J. Royal Statistical Soc. B*, vol. 21, pp. 272-319, 1959.
- [16] R. Kohavi and G.H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273-324, 1997.
- [17] P. Langley, "Selection of Relevant Features in Machine Learning," *Proc. AAAI Fall Symp. Relevance*, pp. 140-144, 1994.
- [18] T. Li, C. Zhang, and M. Ogihara, "A Comparative Study of Feature Selection and Multiclass Classification Methods for Tissue Classification Based on Gene Expression," *Bioinformatics*, vol. 20, no. 15, pp. 2429-2437, 2004.
- [19] Y. Li, C. Campbell, and M. Tipping, "Bayesian Automatic Relevance Determination Algorithms for Classifying Gene Expression Data," *Bioinformatics*, vol. 18, pp. 1332-1339, 2004.
- [20] LYM, <http://genome-www.stanford.edu/lymphoma>, 2008.
- [21] MLL, <http://research.dfc.harvard.edu/korsmeyer/MLL.htm>, 2008.
- [22] R. Marko and K. Igor, "Theoretical and Empirical Analysis of ReliefF and RReliefF," *Machine Learning J.*, pp. 23-69, 2003.
- [23] NCI60, <http://genome-www.stanford.edu/nci60/>, 2008.
- [24] C. Ooi and P. Tan, "Genetic Algorithms Applied to Multi-Class Prediction for the Analysis of Gene Expression Data," *Bioinformatics*, vol. 19, pp. 37-44, 2003.
- [25] H. Peng, F. Long, and C. Ding, "Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226-1238, Aug. 2005.
- [26] K.B. Petersen and M.S. Pedersen, *The Matrix Cookbook*, Version 20051003, 2006.
- [27] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, and T. R. Golub, "Multiclass Cancer Diagnosis Using Tumor Gene Expression Signatures," vol. 98, no. 26, pp. 15149-15154, 2001.
- [28] D.T. Ross, U. Scherf, M.B. Eisen, C.M. Perou, C. Rees, P. Spellmand, V. Iyer, S.S. Jeffrey, M. Van de Rijn, M. Waltham, A. Pergamenschikov, J.C.F. Lee, D. Lashkari, D. Shalon, T.G. Myers, J.N. Weinstein, D. Botstein, and M.P.O. Brown, "Systematic Variation in Gene Expression Patterns in Human Cancer Cell Lines," *Nature Genetics*, vol. 24, pp. 227-235, 2000.
- [29] SRBCT, <http://research.nhgri.nih.gov/microarray/Supplement/>, 2008.
- [30] C. Stein, "Estimation of a Covariance Matrix," *Rietz Lecture, 39th IMS Ann. Meeting*, 1975.
- [31] Y. Su, T.M. Murali, V. Pavlovic, and S. Kasif, "Rankgene: Identification of Diagnostic Genes Based on Expression Data," *Bioinformatics*, <http://genomics10.bu.edu/yangsu/rankgene/>, 2003.
- [32] E.P. Xing, M.I. Jordan, and R.M. Karp, "Feature Selection for High-Dimensional Genomic Microarray Data," *Proc. 18th Int'l Conf. Machine Learning (ICML '01)*, pp. 601-608, 2001.
- [33] E.-J. Yeoh, M.E. Ross, S.A. Shurtleff, W.K. Williams, D. Patel, R. Mahrouz, F.G. Behm, S.C. Raimondi, M.V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C.-H. Pui, W.E. Evans, C. Naeve, L. Wong, and J.R. Downing, "Classification, Subtype Discovery, and Prediction of Outcome in Pediatric Lymphoblastic Leukemia by Gene Expression Profiling," *Cancer Cell*, vol. 1, no. 2, pp. 133-143, 2002.
- [34] K.Y. Yeung, C. Fraley, A. Murua, A.E. Raftery, and W.L. Ruzzo, "Model-Based Clustering and Data Transformations for Gene Expression Data," *Bioinformatics*, vol. 17, no. 10, pp. 977-987, 2001.

- [35] K. Yu, J. Bi, and V. Tresp, "Active Learning via Transductive Experimental Design," *Proc. 23rd Int'l Conf. Machine Learning (ICML '06)*, pp. 1081-1088, 2006.
- [36] L. Yu, H. Liu, and V. Tresp, "Redundancy Based Feature Selection for Microarray Data," *Proc. 10th Int'l Conf. Knowledge Discovery and Data Mining (KDD)*, 2004.



Shenghuo Zhu received the BE degree from Zhejiang University in 1994, the BE degree from Tsinghua University in 1997, and the PhD degree in computer science from the University of Rochester in 2003. He is a research staff member at NEC Laboratories America, Cupertino, California. His primary research interests include information retrieval, probabilistic modeling, machine learning, and data mining.



Dingding Wang received the bachelor's degree from the Department of Computer Science, University of Science and Technology of China, in 2003. She is currently a PhD student in the School of Computer Science, Florida International University, Miami. Her research interests are data mining, information retrieval, and machine learning.



Kai Yu received the PhD degree in computer science from the University of Munich, Germany, in 2004. He worked at Siemens as a senior research scientist during 2004-2006. He is currently a research staff member at NEC Laboratories America, Cupertino, California. His research has been focused on probabilistic modeling, Bayesian inference, Gaussian processes, general machine learning algorithms, and their applications to information retrieval, text mining, recommender systems, clinical data analysis, intrusion detection, and visual recognition. He is the author of more than 40 research papers in international conferences and journals and has served as a program committee member or reviewer for conferences like ICML, NIPS, SIGIR, and ECML.



Tao Li received the PhD degree in computer science from the Department of Computer Science, University of Rochester, in 2004. He is currently an associate professor in the School of Computer Science, Florida International University, Miami. His research interests are data mining, machine learning, information retrieval, and bioinformatics.



Yihong Gong received the BS, MS, and PhD degrees in electronic engineering from the University of Tokyo in 1987, 1989, and 1992, respectively. He then joined the Nanyang Technological University of Singapore, where he worked as an assistant professor in the School of Electrical and Electronic Engineering for four years. From 1996 to 1998, he worked at the Robotics Institute, Carnegie Mellon University, as a project scientist. He was a principal investigator for both the Informedia Digital Video Library Project and the Experience-on-Demand Project funded in multimillion dollars by US NSF, DARPA, NASA, and other government agencies. In 1999, he joined NEC Laboratories America and has been leading the Multimedia Processing Group since then. In 2006, he became the site manager to lead the Cupertino branch of the laboratories. His research interests include multimedia content analysis and machine learning applications. The major research achievements from his group include news video summarization, sports highlight detection, data clustering, and Smart-Catch video surveillance that led to a successful spin-off.

► For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.