Available online at www.sciencedirect.com
SCIENCE DIRECT®

Computational Statistics & Data Analysis 46 (2004) 407–425

COMPUTATIONAL STATISTICS & DATA ANALYSIS

www.elsevier.com/locate/csda

On partial least squares dimension reduction for microarray-based classification: a simulation study

Danh V. Nguyen[a,*], David M. Rocke[b]

[a]Division of Biostatistics, School of Medicine, University of California, Davis, CA 95616, USA
[b]Department of Applied Science, University of California, Davis, CA 95616, USA

Received 3 February 2003; received in revised form 2 August 2003

Abstract

In microarray tumor tissue classification studies, the expressions of thousands of genes (variables) are simultaneously measured across a few tissue samples. Standard statistical methodologies in classification do not work well when the dimension, p, is greater than the sample size, N. One approach to classification problems, when $p \gg N$, is to first apply a dimension reduction method and then perform the classification in the reduced space. In this paper, we study dimension reduction for classification in high dimension based on partial least squares (PLS) and principal components analysis (PCA). In addition, we propose and explore two hybrid-PLS methods for dimension reduction. PLS components are linear combinations of the original predictors, but the weights are nonlinear functions of both the predictors and response variable. This makes it difficult to study the PLS classification methodologies analytically, so, in this paper, we turn to a numerical study using simulation.
© 2003 Elsevier B.V. All rights reserved.

Keywords: DNA microarray; Logistic discrimination; Partial least squares; Principal components analysis

1. Introduction

DNA microarray technology, introduced in 1995–1996, allows the measurement of thousands of gene expression values simultaneously, providing insight into the global

* Corresponding author. Department of Epidemiology and Preventive Medicine, University of California, One Shields Avenue, Davis 956168638, USA. Tel.: +5307546510; fax: +5307523239.
E-mail address: ucdnguyen@ucdavis.edu (D.V. Nguyen).

0167-9473/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2003.08.001

gene expression patterns of cells (tissues) being studied (Lockhart et al. 1996; Schena et al., 1995, 1996). Despite the need for further technological developments with microarray assays (Nguyen et al., 2002d), the approach remains powerful for studying the myriad of transcription-related pathways involved in cellular growth, differentiation, and transformation in various organisms. In particular, the ability to measure thousands of gene expressions simultaneously using DNA microarrays has made it possible to investigate genome-wide objective approaches to molecular cancer classification.

A typical DNA microarray data set in tumor tissue classification studies consists of expression measurements on thousands of genes over a small number of known tumor tissue samples ($p \gg N$). However, many standard statistical methodologies for classification and prediction require more samples than predictors. For example, in regression, $N < p$ leads to an ill-posed problem because the ordinary least squares (OLS) solution is not unique. Another example is Fisher's discriminant analysis, where the covariance matrix is singular when $N < p$.

One approach to this problem of classification in high dimension, encountered with microarray data, is to first apply dimension reduction techniques. After dimension reduction, standard classification/prediction tools, such as linear discriminant analysis (LDA) or logistic discrimination (LD), can be implemented in the reduced subspace. The information retained (i.e., the dimension reduction strategy used in the first step) plays an important role in the subsequent prediction. We have found that dimension reduction via partial least squares (PLS) (Höskuldsson, 1988; Helland, 1988) performs well relative principal components analysis in microarray-based tumor classification studies (Nguyen and Rocke, 2002a,b,c). Although the PLS components used in classification/prediction are linear combinations of the predictor variables, the weights are functions of both the predictors and response variable (see Section 3.2). The structure of PLS weights leads to estimates that are nonlinear functions of $\{y_i\}_{i=1}^{N}$, even in the simple case of linear regression estimates. This makes analytical study of the PLS classification methodology difficult, so, in this paper, we turn to a numerical study using simulation.

Also, PLS was originally designed for continuous response variable(s) and the method is advanced in the field of Chemometrics. Many applications of this kind are available in the *Journal of Chemometrics* (Wiley) and *Chemometrics and Intelligent Laboratory Systems* (Elsevier). See also Martens and Naes (1989). In classification problems, the response is categorical. Here, we focus on binary classification only and study the performance of PLS in this setting. Using singular value decomposition (SVD) we characterize the structure of the PLS weights when the response variable is binary. This derivation of the response-dependence structure of PLS components led us to explore a modified PLS procedure for binary response (PLSM2). In addition, using a formulation of PLS as a sequence of simple linear regressions (SLRs) by Garthwaite (1994), we explore another modification of PLS for binary response variables (PLSM1). We compare the performance of these hybrid-PLS methods (PLSM1 and PLSM2) to the traditional PLS method as well as to PCA when $N \ll p$.

In this paper, we provide a simulation study of dimension reduction methods for logistic classification in high dimension based on (1) PLS, (2) hybrid-PLS (PLSM1 and PLSM2) and (3) PCA. In addition, we also compare these methods to unmodified

PLS for classification using estimates from ordinary least squares on binary response (Section 4).

We briefly review logistic discrimination in Section 2 and then describe dimension reduction methods in Section 3. In this section, the relevant aspects of PCA and PLS are introduced. In addition, we derive two characterizations of how PLS components depend on the response values, leading us to explore two hybrid-PLS methods (PLSM2 and PLSM1). Classification after dimension reduction using logistic discriminant analysis is described in Section 4. The data simulation procedure used to study the performance of these dimension reduction methods and classification is described in Section 5. Results of the simulation study and a discussion of the limitation of the study is described in Section 6. As this is a simulation study, we will exercise care to not over-generalize conclusions.

2. Logistic discrimination in high dimension

The cancer classification problem using microarray gene expression data is to construct a classifier or prediction algorithm that can accurately predict the class of origin of a tumor tissue, y, based on the expression profile of p genes, denoted by $\mathbf{x} = (x_1, \ldots, x_p)'$. The prediction algorithm is trained on N samples of known classification, $\{(y_i, \mathbf{x}_i)\}_{i=1}^{N}$, from N microarray experiments. For example, in case of logistic discrimination/classification, the logit of the conditional class probabilities ($\pi_i = P(Y_i = 1|\mathbf{x}_i)$), is modelled as a linear function of the gene expression values: $\text{logit}(\pi_i) \equiv \log(\pi_i/(1-\pi_i)) = \mathbf{x}_i'\boldsymbol{\beta}$. Thus, the conditional class probabilities are modelled using the logistic functional form, $\pi_i = \{\exp(\mathbf{x}_i'\boldsymbol{\beta})/(1+\exp(\mathbf{x}_i'\boldsymbol{\beta}))\}$ (see McCullagh and Nelder, 1989; Hosmer and Lemeshow, 1986).

In the traditional setting where $N > p$, the predicted conditional class probabilities, $\{\hat{\pi}_i\}_{i=1}^{N}$, are obtained by replacing the parameter, $\boldsymbol{\beta}$, with its maximum likelihood estimate (MLE) $\hat{\boldsymbol{\beta}}$. The LD classifier is $\hat{y}_i = I(\hat{\pi}_i \geqslant c)$, where $0 < c < 1$ is a probability cutoff point and $I(A)$ is indicator function taking value 1 if A is true and 0 otherwise. For example, with the common choice of $c = 0.5$, a sample profile $\mathbf{x}$ is classified/predicted as belonging to group 1 ($\hat{y} = 1$) if $\hat{\pi} \geqslant 1 - \hat{\pi}$, where $\hat{\pi} = \{\exp(\mathbf{x}'\hat{\boldsymbol{\beta}})/(1+\exp(\mathbf{x}'\hat{\boldsymbol{\beta}})\}$.

However, the estimation of π_i (i.e. $\boldsymbol{\beta}$) for classification requires that $N > p$. We use dimension reduction to reduce the dimension p to $K \ll N$ (Section 3). Classification becomes feasible after dimension reduction, where the original sample profiles, $\{\mathbf{x}_i\}_{i=1}^{N}$, are replaced by the corresponding "component" profiles in the reduced subspace (Section 4). More precisely, the matrix of predictor values, $\mathbf{X}$, is approximated by the matrix of components, $\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_K]$. In the next section, we consider dimension reduction methods for obtaining $\mathbf{T}$ based on PLS, hybrid-PLS methods, and PCA.

3. Dimension reduction methods

Principal components analysis is a well-known dimension reduction technique and in this section we only give a brief review of selected aspects of PCA that are related to PLS. After the relevant aspects of PLS are described, we introduce two hybrid-PLS methods for binary responses (PLSM1 and PLSM2), based on modifications of the traditional PLS approach.

3.1. Principal components analysis

Dimension reduction of the p-dimensional space by PCA is achieved by constructing principal components (PCs), which are linear combinations of the original p predictor/explanatory variables. More precisely, in PCA, orthogonal linear combinations are constructed to maximize the variance of the linear combination of the explanatory variables sequentially,

$$\mathbf{w}_k = \underset{\mathbf{w}'\mathbf{w}=1}{\operatorname{argmax}}\, \operatorname{var}(\mathbf{X}\mathbf{w}) = \underset{\mathbf{w}'\mathbf{w}=1}{\operatorname{argmax}}\, (N-1)^{-1}\mathbf{w}'\mathbf{S}\mathbf{w}, \tag{1}$$

subject to the orthogonality constraints $\mathbf{w}_k'\mathbf{S}\mathbf{w}_j = 0$, for all $1 \leqslant j < k$. We have used the notation $\mathbf{S} = \mathbf{X}'\mathbf{X}$ where $\mathbf{X}$ is the $N \times p$ matrix of predictor values. The maximum number of nonzero components is the rank of $\mathbf{X}$. Since $\mathbf{X}$ is assumed to be centered the rank is $N-1$ because $N-1 < p$. The kth step of PCA seeks the strongest mode of variation and the $k-1$ orthogonality constraints imposed require that the kth linear combination identifies a mode of variation distinct from those previously identified (by the previous $k-1$ gene components). For details, the reader is referred to Jolliffe (1986).

For example, consider PCA of the correlation matrix, $\mathbf{R} = (1/(N-1))\mathbf{X}'\mathbf{X}$, for a standardized data matrix $\mathbf{X}$. The principal components are obtained from the spectral decomposition, $\mathbf{R} = \mathbf{V}\boldsymbol{\Delta}\mathbf{V}'$, where $\boldsymbol{\Delta} = \operatorname{diag}\{\lambda_1 \geqslant \cdots \geqslant \lambda_{N-1}\}$, $\{\lambda_k\}_{k=1}^{N-1}$ are the eigenvalues, and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_{N-1})$ are the corresponding eigenvectors. The PCs are constructed as $\xi_k = \mathbf{X}\mathbf{v}_k$.

In PCA of the correlation matrix, the ratio of the explained to total variability has a simple form in terms of the eigenvalues. Since $\operatorname{var}(X_j) = 1$ for all j, the total variability is p. It is straightforward to show that $\operatorname{var}(\xi_k) = \lambda_k$ and $\sum_{j=1}^{p} \operatorname{var}(X_j) = \sum_{k=1}^{N-1} \operatorname{var}(\xi_k) = \sum_{k=1}^{N-1} \lambda_k$ (e.g., see Mardia et al., 1979). Thus, the proportion of variation explained by the kth PC is λ_k/p and the cumulative proportion for K PCs is $\sum_{k=1}^{K} \lambda_k/p$. If there are only d $(< N < p)$ underlying components which explain nearly all of the observed variation then we expect that $\sum_{k=1}^{d} \lambda_k/p \approx 1$.

Prediction/classfication using standard methods can then be carried out in the reduced space by using the constructed PCs. For instance, prediction of a continuous response vector, $\mathbf{y}$, based on the constructed PCs is the well-known principal component regression (PCR) method (Massey, 1965). A PCR model is the linear regression model based on the subspace spanned by K PCs, $\{\xi_1, \ldots, \xi_K\}$.

3.2. Partial least squares

In PCA, dimension reduction is achieved by constructing linear combinations that maximize the variance-based objective function, namely $\operatorname{var}(\mathbf{X}\mathbf{w})$. A parallel formulation can be made for PLS, but with an objective function based on covariance. More precisely, PLS components are linear combinations of the predictor variables, constructed to maximize an objective criterion based on the sample covariance between $\mathbf{y}$ and $\mathbf{Xw}$, namely $\operatorname{cov}(\mathbf{Xw}, \mathbf{y})$. Thus, the kth PLS component is obtained by finding the weight vector, $\mathbf{w}$, satisfying

$$\mathbf{w}_k = \underset{\mathbf{w}'\mathbf{w}=1}{\operatorname{argmax}} \operatorname{cov}(\mathbf{Xw}, \mathbf{y}) = \underset{\mathbf{w}'\mathbf{w}=1}{\operatorname{argmax}} (N-1)^{-1}\mathbf{w}'\mathbf{X}'\mathbf{y}. \tag{2}$$

Similar to PCA, the components, $\mathbf{t}_k = \mathbf{Xw}_k$, are required to be orthogonal: $\mathbf{t}_k'\mathbf{t}_j = \mathbf{w}_k'\mathbf{S}\mathbf{w}_j = 0$ for all $1 \leqslant j < k$. The maximum number of PLS components is at most the rank of $\mathbf{X}$.

Analogous to PCR, a PLS regression model with K PLS components is based on the subspace spanned by the first K PLS components, $\mathbf{T}_K = \{\mathbf{t}_1, \ldots, \mathbf{t}_K\}$. In practice, cross-validation is used to determine the number of dimension, K. In seeking dimension reduction useful for prediction, the objective criterion of PLS may be more sensible than PCA since there is no a priori reason why components with high predictor variation should be strongly related to the response variable. A component with small predictor variance could be a better predictor of the response variable (Jolliffe, 1986).

Although the PLS objective criterion (2) may be appealing because it incorporates both response and predictor information into the dimension reduction process, it does lead to weight vectors, $\{\mathbf{w}_i\}_{i=1}^N$, that are functions of the predictors and the response variable. As mentioned in the Introduction section, PLS was originally designed for continuous response variable(s), although the basic algorithm (see Höskuldsson, 1988; De Jong, 1993) can be used for binary response variables. However, there are obvious drawbacks, similar to using ordinary least squares regression on binary response. In the next section, we characterize the structure of the PLS weights and how they depend on the response values, $\{y_i\}_{i=1}^N$. This characterization leads us to suggest a hybrid-PLS method for binary response variables.

3.3. A hybrid-PLS method based on singular value decomposition

Based on SVD of $\mathbf{X}$, we show that the sequence of PLS weights, $\{\mathbf{w}_k\}_{k=1}^N$, and PLS components, $\{\mathbf{t}_k\}_{k=1}^N$, are linear combinations of the eigenvectors of $\mathbf{X}'\mathbf{X}$ and $\mathbf{XX}'$, respectively. Furthermore, the coefficients of the linear combinations depend on the response values, $\{y_i\}_{i=1}^N$, only through the dot product, $\{a_i\} \equiv \{\mathbf{u}_i \cdot \mathbf{y} = \mathbf{u}_i'\mathbf{y}\}_{i=1}^N$, where $\{\mathbf{u}_k\}_{k=1}^N$ are the eigenvectors of $\mathbf{XX}'$. To state these results more precisely, we define some notations associated with SVD. Proofs of the results are straightforward and deferred to Appendix B.

For a real data matrix $\mathbf{X}$ of size $N \times p$ and $N < p$, the SVD of $\mathbf{X}$ is given by

$$\mathbf{X} = \sum_{i=1}^{N} \lambda_i \mathbf{u}_i \mathbf{v}_i' = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}', \tag{3}$$

where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_N)$, $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_N)$, and $\boldsymbol{\Delta} = \operatorname{diag}\{\lambda_1, \ldots, \lambda_N\}$. The constants $\lambda_1 \geqslant \cdots \geqslant \lambda_N \geqslant 0$ are the singular values of $\mathbf{X}$ and $\{(\lambda_i^2, \mathbf{u}_i)\}_{i=1}^N$ are the eigenvalue–eigenvector pairs of $\mathbf{XX}'$. Also, $\mathbf{v}_i = \lambda_i^{-1}\mathbf{X}'\mathbf{u}_i$ is the ith eigenvector of $\mathbf{X}'\mathbf{X}$ with the same nonzero eigenvalue λ_i^2. Based on the SVD of $\mathbf{X}$, we have the following results. (Note that we are only interested in the case when $p \gg N$, although the results below apply when $N > p$ as well.)

Result 3.1. *The kth PLS component is* $\mathbf{t}_k = \sum_{i=1}^{k} r_{ki}\mathbf{u}^{(2i)}$, *where* $\mathbf{u}^{(s)} = \sum_{i=1}^{N} \lambda_i^s a_i \mathbf{u}_i$ *and the coefficients*, r_{ki}, *are functions of* $b_s = \sum_{i=1}^{N} \lambda_i^s a_i^2$ $(s = 2, 4, \ldots)$.

Result 3.2. *The kth PLS weight vector is* $\mathbf{w}_k = \sum_{i=1}^{N} (\lambda_i - d_1\lambda_i^3 - d_2\lambda_i^5 - d_3\lambda_i^7 - \cdots - d_{k-1}\lambda_i^{2k-1})a_i\mathbf{v}_i$ *where* $d_j = \sum_{i=1}^{k} k_{ij}$ *and* k_{ij} *is a functions of* b_s $(s = 2, 4, \ldots)$.

Note that the PLS weight vector is of the form $\sum_{i=1}^{N} \theta_i \mathbf{v}_i$, where the coefficient, θ_i, depends on the response values only through the dot product $a_i = \mathbf{u}_i'\mathbf{y}$. Thus, modifying the form of a_i, provide alternative ways to incorporate different response-dependence structure. We investigate one such modification of PLS for binary response as follows.

Note that the slope of the simple regression of Y on U_i equals $\mathbf{y}'\mathbf{u}_i/(\mathbf{u}_i'\mathbf{u}_i) = \mathbf{y}'\mathbf{u}_i = a_i$, when $\mathbf{u}_i$ has norm 1. Thus, when $\mathbf{X}$ is centered, the a_i's are the slope coefficients. When Y is binary, application of the standard PLS results in linear regression of a binary response on U_i. There are some drawbacks associated this procedure, as linear regression is designed for a continuous response variable. Thus, we explore a hybrid-PLS algorithm obtained by replacing a_i by the coefficient of the logistic regression (LR) model. The components generated according to this modified PLS algorithm is denoted as PLSM2 throughout the manuscript. In addition, we investigate the performance of another hybrid-PLS algorithm (PLSM1), as will be described in the next section.

3.4. A hybrid-PLS method based on logistic regression predictors

The PLS components can also be obtained as linear combinations of SLR predictors (Garthwaite, 1994). Along this line, we show that the PLS components can be expressed as weighted averages of the original predictor/explanatory variables, with weights depending on the sample predictor variances and the partial correlation coefficients. This formulation also suggests another way to modify PLS for binary response variables.

Define $Y_{(1)} = Y$ and $X_j = X_{(1)j}$, where the subscript in parentheses refers the first (kth) component. The first PLS component is obtained by regressing $Y_{(1)}$ on each of the explanatory variables separately. Regressing $\{Y_{(1)}$ on $X_{(1)j}\}_{j=1}^{p}$ gives p SLR predictors of Y, $\{\hat{Y}_{(1)j} = b_{(1)j}X_{(1)j}\}_{j=1}^{p}$, where $\{b_{(1)j}\}_{j=1}^{p}$ are the SLR slope coefficients. The first PLS component, T_1, is a linear combination of the p SLR predictors

$$T_1 = \sum_{j=1}^{p} v_{(1)j}\hat{Y}_{(1)j} = \sum_{j=1}^{p} v_{(1)j}b_{(1)j}X_{(1)j}, \tag{4}$$

where $v_{(1)j} = (N-1)\,\mathrm{var}\,(X_{(1)j})$. Denoting $s_{(1)y} = \sum_{i=1}^{N}(y_{(1)i} - \bar{y}_{(1)})^2$ and $s_{(1)j} = \sum_{i=1}^{N}(x_{(1)ij} - \bar{x}_{(1)j})^2$, the jth slope coefficient can be expressed as $b_{(1)j} = (s_{(1)y}/s_{(1)j})\,\mathrm{Corr}(X_{(1)j}, Y_{(1)})$, where $\mathrm{Corr}(X_{(1)j}, Y_{(1)})$ is the correlation between $X_{(1)j}$ and $Y_{(1)}$. Thus, from the second equality in (4), it follows that T_1 is a linear combination of the original predictors with coefficients depending on the sample predictor variances and the correlation coefficients.

As described by Garthwaite (1994), the variability in the explanatory variable, X_j, not accounted for by the first PLS component, T_1, can be estimated by the residual variables, denoted $\{X_{(2)j}\}_{j=1}^{p}$, from regressing $X_{(1)j}$ on T_1. Similarly, the variability in response $Y_{(1)}$ not explained by T_1 is estimated by the residual variable, $Y_{(2)}$, from the regressing $Y_{(1)}$ on T_1. The second PLS component, T_2, is constructed from regressing the Y-residual variable on each X-residual variable separately: $\{Y_{(2)}$ on $X_{(2)j}\}_{j=1}^{p}$. Similar to the first PLS component, T_2 is a linear combination of the p SLR predictors (of $Y_{(2)}$),

$$T_2 = \sum_{j=1}^{p} v_{(2)j}\hat{Y}_{(2)j} = \sum_{j=1}^{p} v_{(2)j}b_{(2)j}X_{(2)j} = \sum_{j=1}^{p} v_{(2)j}^{*}\,\mathrm{Corr}_{T_1}(X_{(1)j}, Y_{(1)})X_{(2)j}, \tag{5}$$

where $v_{(2)j} = (N-1)\,\mathrm{var}(X_{(2)j})$. Note that the last equality follows since $b_{(2)j} = (s_{(2)y}/s_{(2)j})\,\mathrm{Corr}_{T_1}(X_{(1)j}, Y_{(1)})$, where $\mathrm{Corr}_{T_1}(X_{(1)j}, Y_{(1)})$ is the partial correlation between $X_{(1)j}$ and $Y_{(1)}$, adjusted for the effect of T_1. The scale factors $\{v_{(2)j}^{*}\}_{j=1}^{p}$ depend on the sample variance of $X_{(2)j}$ and $Y_{(2)}$.

The PLS components for $k \geqslant 2$ are constructed as in the construction of T_2 and is given by

$$T_k = \sum_{j=1}^{p} v_{(k)j}\hat{Y}_{(k)j} = \sum_{j=1}^{p} v_{(k)j}^{*}\,\mathrm{Corr}_{T_{k-1}}(X_{(k-1)j}, Y_{(k-1)})X_{(k)j}, \quad k = 2,\ldots,K, \tag{6}$$

where K is less than or equal to the rank of $\mathbf{X}$. Thus, coefficients of the linear combination depends on the response values, $\{y_i\}_{i=1}^{N}$, through the scaled partial correlations, $\{v_{(k)j}^{*}\,\mathrm{Corr}_{T_{k-1}}(X_{(k-1)j}, Y_{(k-1)})\}_{j=1}^{p}$.

Based on the above construction, various response-dependence structures can be explored by altering the partial correlation structure in (6). For example, when the response variable is binary, alternative correlation measures for a binary response variable may be more appropriate. The above formulation of PLS leads us to explore a direct modification of the PLS components by a sequence of logistic regression predictors, rather than simple linear regression predictors, since the response variable is binary. More precisely, in the original PLS algorithm, the sequence of linear regression predictors is replaced by a corresponding sequence of logistic regression predictors, since Y is binary. We refer to this hybrid-PLS procedure as PLSM1 throughout this manuscript.

4. Classification after dimension reduction

As mentioned earlier, our primary goal is to investigate further dimension reduction methods based on PLS for classification via logistic discrimination. Because our focus here is on the dimension reduction step, we fixed the classification step with LD, although other methodologies can be used (e.g., see Nguyen et al., 2002d). After dimension reduction, the original data matrix is approximated by the matrix of components,

Table 1
Combination of dimension reduction and classification methods examined

	Notation/procedure	Step 1: Dimension reduction	Step 2: Classification	Estimated values
1	PLS–LD	PLS $\rightarrow \mathbf{T}_{pls}$	$\Rightarrow$ LD	$\rightarrow \hat{\mathbf{y}}_{pls}$
2	PLSM2–LD	PLSM2 $\rightarrow \mathbf{T}_{plsm2}$	$\Rightarrow$ LD	$\rightarrow \hat{\mathbf{y}}_{plsm2}$
3	PLSM1–LD	PLSM1 $\rightarrow \mathbf{T}_{plsm1}$	$\Rightarrow$ LD	$\rightarrow \hat{\mathbf{y}}_{plsm1}$
4	PCA–LD	PCA $\rightarrow \mathbf{T}_{pc}$	$\Rightarrow$ LD	$\rightarrow \hat{\mathbf{y}}_{pc}$
5	PLS–OLS	PLS $\rightarrow \mathbf{T}_{pls}$	$\Rightarrow$ OLS	$\rightarrow \hat{\mathbf{y}}_{ols}$

$\mathbf{T} = [\mathbf{t}_1, \ldots, \mathbf{t}_K]$, constructed by PLS, PLSM2, PLSM1, or PCA, as described in the previous section. Thus, the estimated conditional class probability for sample i is

$$\hat{\pi}_i = \{\exp(\mathbf{t}^{(i)}\hat{\boldsymbol{\beta}}^*)\}/\{1 + \exp(\mathbf{t}^{(i)}\hat{\boldsymbol{\beta}}^*)\}, \tag{7}$$

where $\mathbf{t}^{(i)}$ is the ith row of $\mathbf{T}$ and $\hat{\boldsymbol{\beta}}^*$ is the $K \times 1$ vector of MLEs based on $\mathbf{T}$.

Table 1 summarizes the five procedures that we will explore in the simulation study (see Section 5). Algorithmic summaries of the procedures are given in Appendix A. The first four procedures, use dimension reduction via PLS, PLSM2, PLSM1 and PCA and classification using LD. The fifth procedure (denoted PLS–OLS) uses PLS dimension reduction as in the first procedure (PLS–LD), but the second step uses OLS estimates instead. We included this additional procedure in our study because it is essentially unmodified PLS, therefore treating the binary response variable as though it is a continuous response variable. In the next section, we describe the simulation of the data, $\{y_i, \mathbf{x}_i\}_{i=1}^N$, needed for a numerical study of the procedures summarized in Table 1.

Note that we fix the classification method in the simulation study, namely LD. Our interest is to study the dimension reduction methods based on PLS and their performance relative to PCA in high dimension. We have found that the combination of methods, such as PLS–LD, works reasonably well in applications to real microarray data (e.g., Nguyen et al., 2002d). However, we point out here that many discriminant analysis methodologies have been proposed for classification based on gene expression data. The latest discriminant analysis techniques for microarray data can be found in Dudoit et al. (2002), Furey et al. (2000) and Ben-Dor et al. (2000) among others. Our focus here is on studying the dimension reduction step and not on comparing classification methodologies, which has already been carried out (Dudoit et al., 2002; Ben-Dor et al., 2000 among others).

5. Simulation procedure

The first few principal components of real microarray data can explain a wide range of variability in the data. Thus, we design a flexible simulation procedure for generating data, $\{y_i, \mathbf{x}_i\}_{i=1}^N$, so that the first K PCs explain a specified proportion of predictor variability. The amount of variability explained is then allowed to vary in a wide range to encompass most situations encountered in practice (30–90%). From the generated

Table 2
Summary and notation of simulation model

		Model	Parameters
Components	$\{\boldsymbol{\tau}_k\}_{k=1}^{d}$	$\mathrm{N}(\mu_\tau, \sigma_\tau^2)$	μ_τ, σ_τ^2
Error	$\{\boldsymbol{\varepsilon}_i\}_{i=1}^{N}$	$\mathrm{N}(\mu_\varepsilon, \sigma_\varepsilon^2)$	$\mu_\varepsilon, \sigma_\varepsilon^2$
Data matrix	$\{x_{ij}\}_{i,j=1}^{N,p}$	$\mathrm{LN}(a_i, b_i^2)$	$a_i = \mu_\tau \sum_{k=1}^{d} r_{ki},\ b_i^2 = \sigma_\tau^2 \sum_{k=1}^{d} r_{ki}^2 + \sigma_\varepsilon^2$

data matrix, $\mathbf{X}$, the conditional class probabilities are obtained as $\pi_i = P(Y_i = 1|\mathbf{x}_i) = \exp(\mathbf{x}_i'\boldsymbol{\beta})/(1+\exp(\mathbf{x}_i'\boldsymbol{\beta}))$ and the binary response values are generated as $Y_i \sim \mathrm{Bin}(1, \pi_i)$. Details of the simulation procedure follow.

5.1. Generating the data matrix of predictor values

The ith sample (row) of the $N \times p$ data matrix is generated from a basic model with d underlying components

$$\mathbf{x}_i^* = r_{1i}\boldsymbol{\tau}_1 + \cdots + r_{di}\boldsymbol{\tau}_d + \boldsymbol{\varepsilon}_i, \quad i = 1, \ldots, N. \tag{8}$$

More precisely, $x_{ij}^* = \sum_{k=1}^{d} r_{ki}\tau_{kj} + \varepsilon_{ij}$, where $\{\boldsymbol{\tau}_k = (\tau_{k1}, \ldots, \tau_{kp})'\}_{k=1}^{d}$ are the components, $\{\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \ldots, \varepsilon_{ip})'\}_{i=1}^{N}$ are i.i.d. vectors of noise, and $\{r_{1i}, \ldots, r_{di}\}$ is a set of fixed constants. We take the component values as $\tau_{kj} \sim \mathrm{N}(\mu_\tau, \sigma_\tau^2)$ and the noise values as $\varepsilon_{ij} \sim \mathrm{N}(\mu_\varepsilon, \sigma_\varepsilon^2)$. The matrix of predictor values are obtained as $x_{ij} = \exp(x_{ij}^*)$. Note that $x_{ij}^* \sim \mathrm{N}(\mu_\tau \sum_{k=1}^{d} r_{ki}, \sigma_\tau^2 \sum_{k=1}^{d} r_{ki}^2 + \sigma_\varepsilon^2) \equiv \mathrm{N}(a_i, b_i^2)$. Thus, x_{ij} is distributed as log-normal with parameters a_i and b_i^2, denoted as $x_{ij} \sim \mathrm{LN}(a_i, b_i^2)$. A log-normal model has been used for microarray expression data (e.g., see Rocke and Durbin, 2001). Data values generated according to (8) suggest that each row of $\mathbf{X}$ comprises of a linear combination of d underlying components and a random error component. Table 2 summarizes the simulation procedure for the predictor data matrix.

The data matrix is generated with a mean noise of zero, $\mu_\varepsilon = 0$, and with a component mean of $\mu_\tau = 5/d$. The number of components is fixed at $d = 6$. The relative variance between the noise and component factors is controlled by the ratio of variance parameter $\delta = \sigma_\varepsilon/\sigma_\tau$. Thus, the ability to separate the noise from the component signal is controlled by varying δ. For example, the separation between noise and component signal can be decreased by increasing the noise variance parameter (σ_ε) for a fixed σ_τ.

5.2. Generating the conditional class probabilities

The binary response values, $\{y_i\}_{i=1}^{N}$, are generated as $Y_i \sim \mathrm{Bin}(1, \pi_i)$, where $\pi_i = \{\exp(\mathbf{x}_i'\boldsymbol{\beta})\}/\{1 + \exp(\mathbf{x}_i'\boldsymbol{\beta})\}$ and $\boldsymbol{\beta}$ a fixed vector of parameters. A careful choice of $\boldsymbol{\beta}$ is necessarily to avoid producing too many π_i close to zero or one, leading to data configurations of complete or quasicomplete separation (Albert and Anderson, 1984). Maximum likelihood estimates do not exist for these data configurations. Thus, it is important to ensure that the conditional class probabilities, $\{\pi_i\}_{i=1}^{N}$, generated do not cluster at zero and one, as classification for such data sets will be too easy.

Table 3
For each combination of simulation parameters $(p, ave(\lambda, 3), \delta, \sigma_\pi)$ 100 data sets are generated. The proportion of variability explained by K principal components is given by $ave(\lambda, K) = \sum_{k=1}^{K} \lambda_k/p$

		p							
		100	300	500	800	1000	1200	1400	1600
$ave(\lambda, 3)$	$\delta = \sigma_\varepsilon/\sigma_\tau$	σ_π							
33%	0.01	0.280	0.150	0.115	0.090	0.080	0.073	0.0705	0.0700
50%	0.05	0.282	0.151	0.117	0.091	0.088	0.079	0.0702	0.0701
72%	0.10	0.350	0.200	0.130	0.105	0.095	0.080	0.0790	0.0780
90%	0.20	0.305	0.205	0.135	0.110	0.105	0.081	0.0791	0.0780

To avoid such data configurations, we control the magnitude of $\{\beta_j\}_{j=1}^{p}$ (in conjunction with δ), using the following simple procedure. Because the number of predictor variables, p, is large (ranging from 100 to 1600), we find it convenient to select them from an $N(0, \sigma_\pi^2)$ distribution. The simulation parameter σ_π^2 allows overall control over the selection of $\{\beta_j\}_{j=1}^{p}$. For a given value of p, ranging from 100 to 1600, Table 3 lists the value of σ_π which gives $\{\pi_i\}_{i=1}^{N}$ about evenly distributed on (0,1).

5.3. Simulation size and parameter settings

Note that the data matrix is of size $N \times p$, where $p \gg N$. Because this is the case of interest in practice, we consider the number of predictors in the range of 100–1600, with the sample size, N, fixed at a small value of 40. Although the dimension of microarray data, p, in practice is in the thousands, often a smaller subset of hundreds are selected for analysis (see e.g., Nguyen and Rocke, 2002b; Ambroise and McLachlan, 2002).

As mentioned earlier, a predictor data matrix is generated so that the first K PCs explain a specified proportion of predictor variability. This is controlled by the simulation parameter δ. Data sets are generated so that the proportion of total variability explained by the first K PCs, namely $ave(\lambda, K) = \sum_{k=1}^{N} \lambda_k/p$, is between 0.30 and 0.90 (see Table 3). The number of dimension retained, K, should be fixed across methods and we fixed it to be 3 in the simulation. For each percentage of variability explained, $100 \times ave(\lambda, K) \in \{32\%, 50\%, 72\%, 90\%\}$, and for each $p = 100$, 300, 500, 800, 1000, 1200, 1400, and 1600, one hundred data sets were generated. The exact simulation parameter settings are given in Table 3.

6. Results and discussion

6.1. Simulation results

The estimated conditional class probabilities, $\{\hat{\pi}\}_{i=1}^{N}$, were obtained for each simulated data set using logistic regression based on $K = 3$ components (Section 4). The

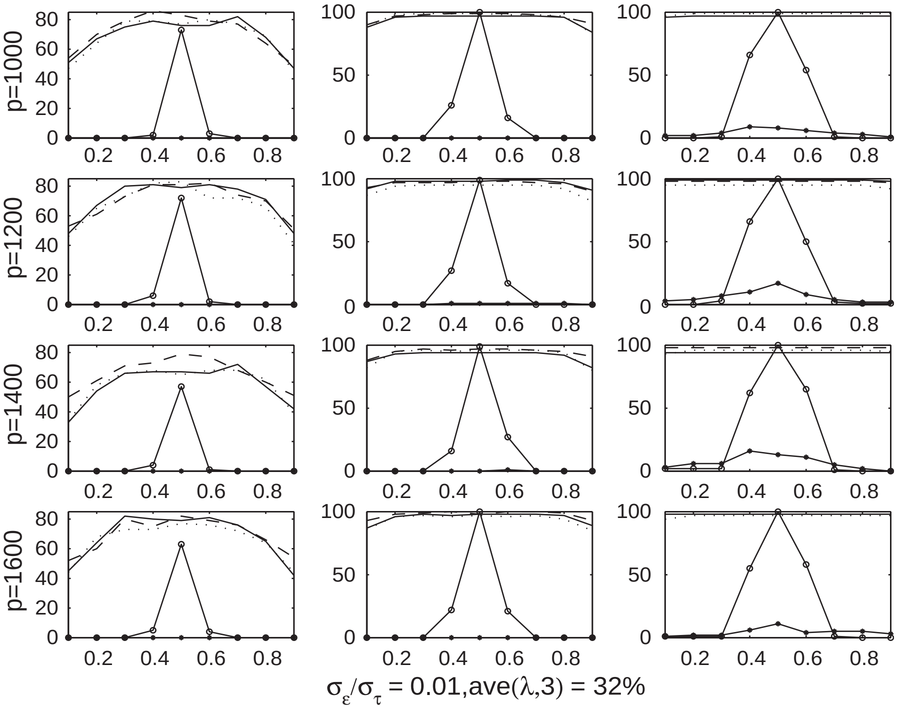

Fig. 1. Percentage of correct classification for data sets generated with 32% predictor variability explained. The y-axis of each plot is the percentage of data sets, out of 100, with correct classification $\geqslant 90\%$ (column 1), $\geqslant 80\%$ (column 2) , and $\geqslant 70\%$ (column 3). Results are given as a function of the cutoff point c (x-axis) and for each dimension $p = 1000$, 1200, 1400, and 1600. The methods are PCA–LD ($- * -$), PLS–LD (——), PLSM2–LD ($\cdots$), PLSM1–LD ($- - -$), PLS–OLS($- \circ -$). Results for $p = 100$, 300, 500, and 800 are similar and are given at the supplemental web site http://dnguyen.ucdavis.edu/.html/supplemental.html.

components were obtained using PLS, PLSM2, PLSM1, and PCA (see Table 1). In addition, the unmodified procedure of OLS on binary responses rather than LR using PLS components, namely PLS–OLS, was also applied as a baseline comparison. Classification of sample i is $\hat{y}_i = I(\hat{\pi}_i \geqslant c)$, where c is a pre-specified probability cutoff and $c \in \{0.1, 0.2, \ldots, 0.9\}$. Note that specifying $c = 0.5$ corresponds to the common procedure of assigning $\hat{y}_i = 1$ for $\hat{\pi}_i \geqslant 1 - \hat{\pi}_i$. The proportion of correct classification for any given data set is $\#\{\hat{y}_i = y_i\}/N$.

The performance of the five procedures, PLS–LD, PLSM2–LD, PLSM1–LD, PCA–LD, and PLS–OLS, across 800 simulated data sets is summarized in Fig. 1. One hundred data sets were generated for each dimension $p = 100$, 300, 500, 800, 1000, 1200, 1400, and 1600. For each plot in Fig. 1, the y-axis is the percentage of the data sets, out of 100 generated, which yielded at least 90% correct classification (first column), 80% (second column) or 70% (third column). The x-axis is the cutoff point $c \in \{0.1, 0.2, \ldots, 0.9\}$. The amount of total variability explained by the first $K = 3$

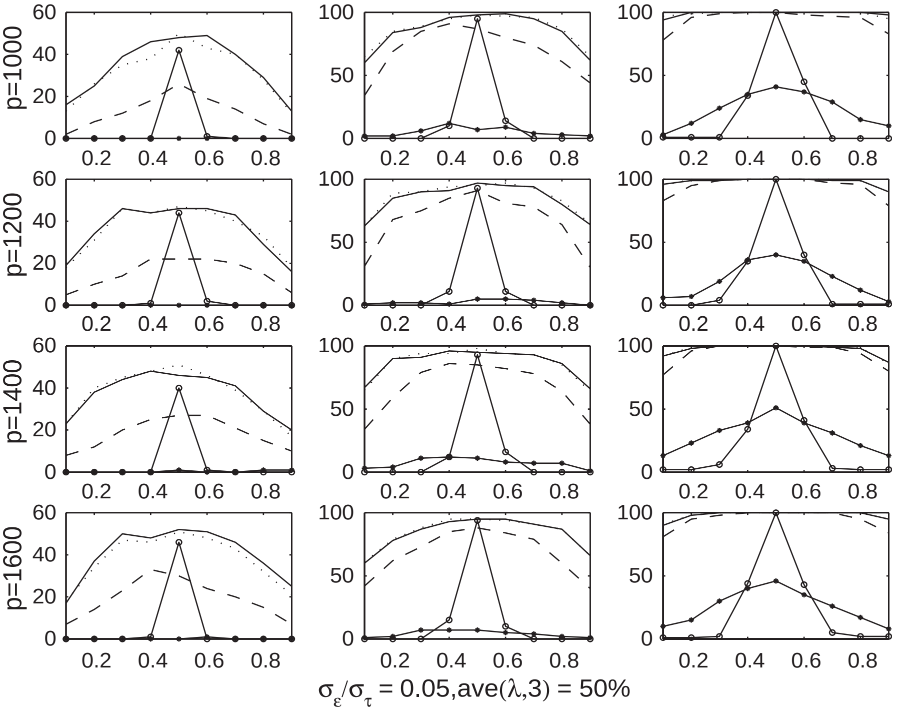

Fig. 2. Percentage of correct classification for data sets generated with 50% predictor variability explained. The methods are PCA–LD ($- * -$), PLS–LD (——), PLSM2–LD (···), PLSM1–LD ($- - -$), PLS–OLS($- \circ -$). See Fig. 1 caption for details.

PCs for each data set is about 32% in Fig. 1. Since the pattern of results are similar for $p = 100$, 300, 500, and 800 the results were not displayed in Fig. 1 (and in subsequent figures). Similarly, Figs. 2 and 3 summarize the performance of the five procedures when the amount of variability explained for each data set increased to 50% and 72%, respectively. (This pattern holds for the 90% variability explained case as well so the results were not displayed.)

When the first three PCs account for 32% of total predictor variability (Fig. 1) classification using PLS components or hybrid-PLS components, namely PLS–LD, PLSM2–LD and PLSM1–LD, performed well: At least 90% correct classification in about 20 –80 out of the 100 data sets (column 1, Fig. 1) was observed. Furthermore, PLS–LD, PLSM2–LD, and PLSM1–LD had at least 80% correct classification in nearly all data sets generated (column 2, Fig. 1). Classification based on principal components did relatively poorly, with only less than 4 of 100 results with at least 80% correct classification (see PCA–LD curves: $- * -$ in column 2 of Fig. 1). In addition, PCA–LD resulted in none with at least 90% correct (the flat $- * -$ lines at zero in column 1 of Fig. 1).

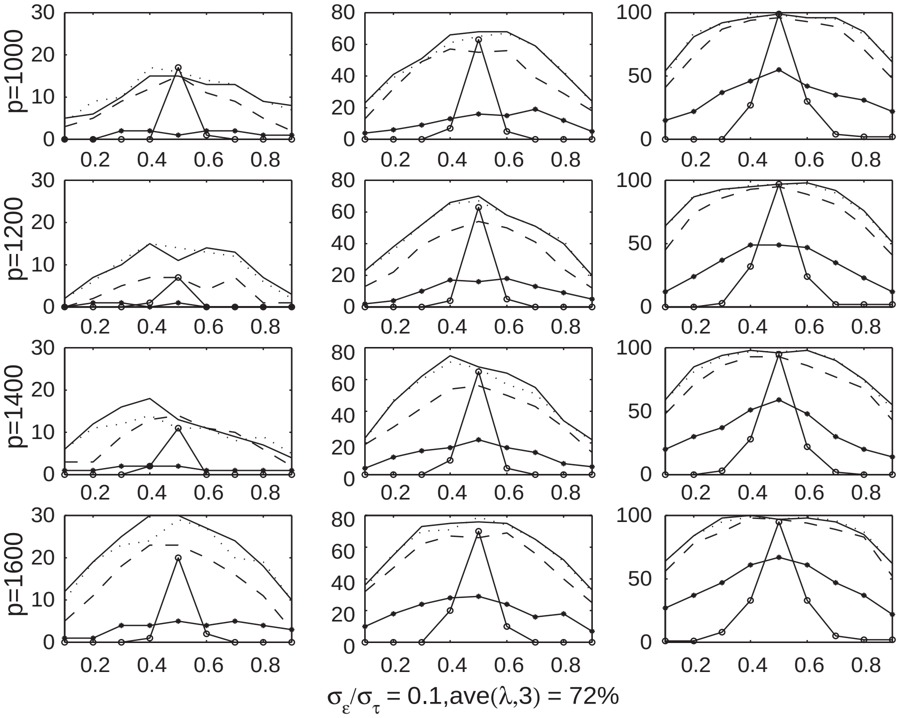

Fig. 3. Percentage of correct classification for data sets generated with 72% predictor variability explained. The methods are PCA–LD (− ∗ −), PLS–LD (——), PLSM2–LD (· · ·), PLSM1–LD (− − −), PLS–OLS(− ○ −). See Fig. 1 caption for details.

The relatively poor performance resulting from classification with principal components (in Fig. 1) is not surprising, because the information used in prediction only captured less than $\frac{1}{3}$ of the total variability in the data. Thus, as expected, classification based on principal components did improve as the proportion of total predictor variability increases to 50% (Fig. 2), 72% (Fig. 3), and 90% (not shown). However, the improvement of PCA–LD fell far below the performance of PLS, PLSM2, and PLSM1 for the same data sets. For example, with the predictor variability explained increasing to approximately 72%, only 20 of 100 data sets have at least 80% correct classification on average (Fig. 2, column 2). Note that in practice, a principal component (dimension) that explain only a small proportion of total variability can be a very good predictor (see Section 6.2). However, no such dimension were assigned to be of more predictive value, a priori, in the simulation.

The classification performance using PCA for dimension reduction depends heavily on the total predictor variability explained (in this simulation). This is reflective of the variance objective function, var(**Xw**), used in the PCA dimension reduction step. On the other hand, the classification performance patterns of PLS remain similar as the total explained predictor variability increased. This is reflective of the covariance objective function of PLS, which requires optimizing both predictor variance and correlation with the response variable.

The unmodified PLS procedure, namely PLS–OLS based on PLS components, did poorly when c is not close to 0.5. The classification performance of PLS–OLS is similar to the other PLS and hybrid-PLS methods at $c = 0.5$, as indicated by the peaks of the PLS–OLS curves (the $- \circ -$ curves in Figs. 1–3). In addition, the classification performance pattern for PLS–OLS remains similar as the percentage of total variability increased. This is expected since the dimension reduction procedure used is exactly PLS. Thus, the performance does not depend solely on the predictor variance.

In summary, the dominant methods in this simulation are PLS–LD (logistic classification using ordinary PLS components) and PLSM2–LD (modification of PLS using the SVD as given in Results 3.1 and 3.2). PLS–OLS using PLS components, which is essentially unmodified PLS, performs satisfactorily with the cutoff point $c = 0.5$, but is still dominated by PLS–LD and PLSM2–LD. The modification of PLS that uses a sequence of logistic regression predictors (PLSM1–LD) is dominated when the variance explained is high. Finally, PCR is not competitive with any of the other methods under this simulation, except for PLS–OLS with cutoff points c away from 0.5.

The classification performance patterns for principal components relative to PLS components in the simulation are similar to results from real gene expression data (see, for example, Nguyen and Rocke, 2002a,b,c). The results of the simulation studies provide some support for the use of PLS as a dimension reduction for classification in high dimension, such as microarray gene expression data. However, as we will discuss in the next section, there are some limitations to the simulation studies and also some practical issues to consider.

6.2. *Discussion and limitations of the simulation study*

The simulation results, described in the previous section, are only valid for data sets generated according to the model in Table 2 and the parameter settings given in Table 3. The motivation for our simulation design is based on our experience with microarray-based cancer classification studies, where noise in the data is not a negligible factor. However, it is not possible to realistically model (simulate) the global dependence structure in microarray data, which is compounded with a complex noise structure. This is because there are potentially hundreds or thousands of genes expressed at various levels, with each gene affecting many others in coordinated pathways, and the resulting complexity of the gene dependence structure/interaction is enormous. In the simulation design, we have attempted to control the information between noise and component signal and then study the performance of the methods under data scenarios with different amount of predictor variability explained. This is a simplistic model, and more complex models is needed to accurately reflect real microarray gene expression data.

Despite this limitation, we believe that the simulation results are still instructive and support empirical findings from classification studies on real microarray data, reported in Nguyen and Rocke (2002a,b). In particular, we have found that PLS out-performs PCA with microarray data. However, PCA can be quite competitive if one pre-selects the predictors (genes) which are predictive of the response classes before applying PCA. However, this simple variable-filtering strategy can induce severe bias on the estimate of accuracy (Ambroise and McLachlan, 2002; Nguyen and Rocke, 2002b). Alternatively, one can select the PCs with low predictor variation explained, but which predicts well (Jolliffe, 1986). Note that this can be argued as one advantage of PLS over PCA. That is, PLS only involves choosing the number of gene components K whereas PCA entails deciding which K gene components to select.

Also, in the simulation we generated data from a 6-component model and then analyzed data from a 3-component model for each procedure. This allowed a comparison between the dimension reduction methods that is not confounded by different model dimensions. However, as we mentioned earlier, cross-validation is used in practice to choose the number of dimensions K. Clearly, different dimension reduction methods lead to different choices of the number of dimensions and this will affect the subsequent classification. Our experiences with using PLS and PCA for microarrray gene expression data indicate that when the response class is very difficult to separate cross-validation can result in a different K for different methods. The difference between PLS and PCA is usually 2 versus 3 components. However, the classification error from models selected with the (different) optimal K via cross-validation is usually lower for PLS relative to PCA. Also, the optimal K for PLS and PCA, chosen by a validation procedure, is usually the same when the expression patterns between groups are more easily separated. This occurs, for example, when classifying normal and tumor tissues. In addition, from our experiences the cross-validation choice of K does not exceed 4 for both PLS and PCA for gene expression data.

Acknowledgements

We are grateful to two reviewers for their very helpful comments. The research reported in this paper was supported by the National Cancer Institute (CA-90301), National Science Foundation (ACI96-19020 and DMS 98-70172) and National Institute of Environmental Health Sciences/National Institutes of Health (P42ES04699).

Appendix A. Algorithmic summaries, computation and software

There are two basic algorithms for PLS called (standard) PLS (see Höskuldsson, 1988; Helland, 1988; Phatak and De Jong, 1997) and SIMPLS (see De Jong, 1993). When there is one response variable (Y), which is the case we considered in this paper, the two algorithms (PLS1 and SIMPLS) give identical components. We refer the reader to the above references for these algorithms. However, for more than one response variable the algorithms give slightly different components. See De Jong (1993) for details. As PLS–OLS and PCA–LD are standard, we provide below algorithmic summaries of PLSM2–LD, PLSM1–LD, and PLS–LD.

A.1. PLSM2–LD

(1) *Step* 1: Compute PLSM2 components.

  (a) Compute SVD of data matrix $\mathbf{X}_{N\times p} = \mathbf{U}\boldsymbol{\Delta}\mathbf{V}'$.
   (Save eigenvalue/vector pairs $\{(\lambda_i^2, \mathbf{u}_i)\}_{i=1}^{N}$.)

  (b) Compute slopes of logistic regression of Y on U_i: $\{a_i\}$.

  (c) Compute PLSM2 components: $\mathbf{t}_k = \sum_{i=1}^{k} r_{ki}\mathbf{u}^{(2i)}$.
   (The r_{ki}'s (see Appendix B) are functions of $\{(\lambda_i^2, \mathbf{u}_i)\}$ and $\{a_i\}$ from part (1a) and (1b), respectively. Note that only $k = 1,\dots,4(=K)$ are available in Appendix B.)

(2) *Step* 2: Classification.
  Perform LR of Y on $\mathbf{T} = [\mathbf{t}_1,\dots,\mathbf{t}_K]$ and compute conditional class probabilities $\{\hat{\pi}_i\}$ (Section 4). Make prediction/classification as $\hat{y}_i = I(\hat{\pi}_i \geqslant c)$ for cutoff $c \in (0,1)$.

A.2. PLSM1–LD

(1) *Step* 1: Compute PLSM1 components.

  (a) Perform LR of Y $(Y_{(1)})$ on $X_j = X_{(1)j}$, $j = 1,\dots,p \rightarrow \hat{Y}_{(1)}$. Compute T_1 from Eq. (4).

  (b) Perform SLR of T_1 on $X_{(1)j}$, $j=1,\dots,p \rightarrow \hat{X}_{(1)j}$. Compute $X_{(2)j}=X_{(1)j}-\hat{X}_{(1)j}$.

  (c) Perform LR of Y on $T_1 \rightarrow \hat{Y}_{(1)}$, $Y_{(2)} = Y_{(1)} - \hat{Y}_{(1)}$.

  (d) Perform SLR of $Y_{(2)}$ on $X_{(2)j}$, $j = 1,\dots,p \rightarrow \hat{Y}_{(2)j}$. Compute T_2 from Eq. (5).

  (e) For T_k, $k \geqslant 3$, compute similarly as in (1b)–(1d) and compute T_k from Eq. (6).

(2) *Step* 2: Classification.
  Repeat logistic classification exactly as in the PLSM2–LD algorithm, but using PLSM1 components $(T_1,\dots,T_K)$ from Step 1.

A.3. PLS–LD

(1) *Step* 1: Compute PLS components.

  (a) Obtain K PLS components, $\mathbf{T}=[\mathbf{t}_1,\dots,\mathbf{t}_K]$, using PLS1 or SIMPLS algorithm.

(2) *Step* 2: Classification.
  Repeat logistic classification exactly as in the PLSM2–LD algorithm, except use PLS components from step 1 instead.

The simulation study was performed using the software Matlab.

Appendix B. Proofs

B.1. Proofs of Results 3.1 and 3.2

First, we set the following definitions: (D1) $a_i = \mathbf{u}_i'\mathbf{y}$, (D2) $b_s = \sum_{i=1}^{N} \lambda_i^s a_i^2$, for $s = 2, 4, \ldots$ and (D3) $\mathbf{u}^{(s)} = \sum_{i=1}^{N} \lambda_i^s a_i \mathbf{u}_i$ and $\mathbf{u}^{(t)} = \sum_{i=1}^{N} \lambda_i^t a_i \mathbf{u}_i$ for $s, t = 2, 4, \ldots$. From the orthogonality of $\{\mathbf{u}_i\}_{i=1}^{N}$ and definitions (D1)–(D3) the following properties follow: (P1) $\mathbf{u}^{(s)'}\mathbf{u}^{(t)} = b_{s+t}$, (P2) $\mathbf{u}^{(s)'}\mathbf{y} = b_s$, (P3) $\mathbf{XX}'\mathbf{y} = \mathbf{u}^{(2)}$ and (P4) $\mathbf{XX}'\mathbf{u}^{(s)} = \mathbf{u}^{(s+2)}$ for $s, t = 2, 4, \ldots$. From (P1)–(P4) and some tedious, but simple, algebra we have that the kth PLS component is given by $\mathbf{t}_k = \sum_{i=1}^{k} r_{ki}\mathbf{u}^{(2i)}$ with coefficients:

$$r_{11} = 1,$$

$$r_{21} = f_2 b_6 / g_2^2,$$

$$r_{22} = f_2/g_2 \quad \text{where } f_2 = b_2 \text{ and } g_2 = -b_4;$$

$$r_{31} = f_3(b_8^2 - b_{10}b_6)/g_3^2$$

$$r_{32} = f_3(b_{10}b_4 - b_6b_8)/g_3^2$$

$$r_{33} = f_3/g_3 \quad \text{where } f_3 = (b_4^2 - b_2b_6) \text{ and } g_3 = -(b_4b_8 - b_6^2);$$

$$r_{41} = f_4(b_{10}^3 + b_{12}^2 b_6 + b_{14}b_8^2 - b_{10}(b_{14}b_6 + 2b_{12}b_8))/g_4^2$$

$$r_{42} = f_4(-(b_{12}^2 b_4) + b_{10}b_{14}b_4 - b_{10}^2 b_8 - b_{14}b_6b_8 + b_{12}(b_{10}b_6 + b_8^2))/g_4^2$$

$$r_{43} = f_4(-(b_{10}^2 b_6) - b_{12}b_6b_8 + b_{14}(b_6^2 - b_4b_8) + b_{10}(b_{12}b_4 + b_8^2))/g_4^2$$

$$r_{44} = f_4/g_4 \quad \text{where } f_4 = (b_6^3 + b_{10}(b_4^2 - b_2b_6) - 2b - 4b_6b_8 + b_2b_8^2) \text{ and}$$

$$g_4 = -(b_{10}^2 b_4 - 2b_{10}b_6b_8 + b_8^3 + b_{12}(b_6^2 - b_4b_8)).$$

We only present the results for $K=4$ here but higher-order components are similar. This proves Result 3.1 and the proof of Result 3.2 is similar. That is, from the orthogonality of $\{\mathbf{v}_i\}_{i=1}^{N}$ the coefficients of Theorem 3.2 are

$$k_{11y} = b_2/b_4,$$

$$k_{21y} = (b_6/b_4)(f_2/g_2),$$

$$k_{22y} = (f_2/g_2) \quad \text{where } f_2 = (-b_4^2 + b_2b_6) \text{ and } g_2 = (-b_6^2 + b_4b_8);$$

$$k_{31y} = ((-b_{10}b_6 + b_8^2)/h_3)(f_3/g_3),$$

$$k_{32y} = ((b_{10}b_4 - b_6b_8)/h_3)(f_3/g_3),$$

$$k_{33y} = (f_3/g_3) \quad \text{where } f_3 = (b_6^3 + b_{10}(b_4^2 - b_2b_6) - 2b_4b_6b_8 + b_2b_8^2),$$

$$g_3 = (b_{10}^2 b_4 - 2b_{10}b_6b_8 + b_8^3 + b_{12}(b_6^2 - b_4b_8)) \text{ and } h_3 = (b_6^2 - b_4b_8);$$

$$k_{41y} = -((b_{10}^3 + b_{12}^2 b_6 + b_{14} b_8^2 - b_{10}(b_{14} b_6 + 2b_{12} b_8))/h_4)(f_4/g_4)$$

$$k_{42y} = ((b_{12}^2 b_4 - b_{10} b_{14} b_4 + b_{10}^2 b_8 + b_{14} b_6 b_8 - b_{12}(b_{10} b_6 + b_8^2))/h_4)(f_4/g_4)$$

$$k_{43y} = ((b_{10}^2 b_6 - b_{14} b_6^2 + b_{14} b_4 b_8 + b_{12} b_6 b_8 - b_{10}(b_{12} b_4 + b_8^2))/h_4)(f_4/g_4)$$

$$k_{44y} = (f_4/g_4) \quad \text{where}$$

$$f_4 = (b_{10}^3 b_2 + b_{14} b_6^3 + b_{12}^2(-b_4^2 + b_2 b_6) - 2b_{14} b_4 b_6 b_8 + b_{14} b_2 b_8^2 - b_8^4 + 2b_{12} b_8(-b_6^2 + b_4 b_8) - b_{10}^2(b_6^2 + 2b_4 b_8) + b_{10}(2b_{12} b_4 b_6 + b_{14}(b_4^2 - b_2 b_6) - 2b_{12} b_2 b_8 + 3b_6 b_8^2)),$$

$$g_4 = (b_{10}^4 - b_{12}^3 b_4 + b_{14}^2 b_6^2 - b)12 b_{16} b_6^2 - b_{14}^2 b_4 b_8 + b_{12} b_{16} b_4 b_8 - 2b_{12} b_{14} b_6 b_8 + b_{12}^2 b_8^2 - b_{16} b_8^3 - b_{10}^2(b_{16} b_4 + 2b_{14} b_6 + 3b_{12} b_8) + 2b_{10}(b_{12} b_{14} b_4 + b_{12}^2 b_6 + b_8(b_{16} b_6 + b_{14} b_8))),$$

$$h_4 = (b_{10}^2 b_4 - 2b_{10} b_6 b_8 + b_8^3 + b_{12}(b_6^2 - b_4 b_8)).$$

References

Albert, A., Anderson, J.A., 1984. On the existence of maximum likelihood estimates in logistic models. Biometrika 71, 1–10.

Ambroise, C., McLachlan, G.J., 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. Proc. Natl. Acad. Sci. USA 99, 6562–6566.

Ben-Dor, A., Bruhn, L., Friednam, N., Nachman, I., Schummer, M., Yakhini, Z., 2000. Tissue classification with gene expression profiles. J. Comput. Biol. 7, 559–584.

De Jong, S., 1993. SIMPLS: An alternative approach to partial least squares regression. Chemometr. Intell. Lab. Systems 18, 251–263.

Dudoit, S., Fridlyand, J., Speed, T.P., 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. J. Amer. Statist. Assoc. 97, 77–87.

Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., Haussler, D., 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. Bioinformatics 16, 906–914.

Garthwaite, P.H., 1994. An interpretation of partial least squares. J. Amer. Statist. Assoc. 89, 122–127.

Helland, I.S., 1988. On the structure of partial least squares. Comm. Statist. Simulation Comput. 17, 581–607.

Höskuldsson, A., 1988. PLS regression methods. J. Chemometr. 2, 211–228.

Hosmer, D.W., Lemeshow, S., 1986. Applied Logistic Regression. Wiley, New York.

Jolliffe, I.T., 1986. Principal Component Analysis. Springer, New York.

Lockhart, D.J., Dong, H., Byrne, M.C., Follettie, M.T., Gallo, M.V., Chee, M.S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., Brown, E.L., 1996. Expression monitoring by hybridization to high-density oligonucleotide arrays. Nat. Biotechnol. 14, 1675–1680.

Mardia, K.V., Kent, J.T., Bibby, J.M., 1979. Multivariate Analysis. Academic Press, San Diego.

Martens, H., Naes, T., 1989. Multivariate Calibration. Wiley, New York.

Massey, W.F., 1965. Principal components regression in exploratory statistical research. J. Amer. Statist. Assoc. 60, 234–246.

McCullagh, P., Nelder, J.A., 1989. Generalized Linear Models. Chapman & Hall, London.

Nguyen, D.V., Rocke, D.M., 2002a. Tumor classification by partial least squares using microarray gene expression data. Bioinformatics 18, 39–50.

Nguyen, D.V., Rocke, D.M., 2002b. Multi-class cancer classification via partial least squares using gene expression profiles. Bioinformatics 18, 1216–1226.

Nguyen, D.V., Rocke, D.M., 2002c. Classification of acute leukemia based on DNA microarray gene expressions using partial least squares. In: Lin, S.M., Johnson, K.F. (Eds.), Methods of Microarray Data Analysis. Kluwer, Dordrecht, pp. 109–124.

Nguyen, D.V., Arpat, A.B., Wang, N., Carroll, R.J., 2002d. DNA microarray experiments: biological and technological aspects. Biometrics 58, 701–717.

Phatak, A., De Jong, S., 1997. The geometry of partial least squares. J. Chemometr. 11, 311–338.

Rocke, D.M., Durbin, B., 2001. A model for measurement error for gene expression arrays. J. Comput. Biol. 8, 557–569.

Schena, M., Shalon, D., Davis, R.W., Brown, P.O., 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270, 467–470.

Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P.O., Davis, R.W., 1996. Parallel human genome analysis: microarray-based expression monitoring of 1000 genes. Proc. Natl. Acad. Sci. USA 93, 10614–10619.