Sparse Partial Least Squares Classification for High Dimensional Data

Dongjun Chung and Sündüz Keleş

Abstract

Partial least squares (PLS) is a well known dimension reduction method which has been recently adapted for high dimensional classification problems. We develop sparse versions of the recently proposed two PLS-based classification methods using sparse partial least squares (SPLS). These sparse versions aim to achieve variable selection and dimension reduction simultaneously. We consider both binary and multicategory classification. We provide analytical and simulation-based insights about the variable selection properties of these approaches and benchmark them on well known publicly available datasets that involve tumor classification with high dimensional gene expression data. We show that incorporation of SPLS into a generalized linear model (GLM) framework provides higher sensitivity in variable selection for multicategory classification with unbalanced sample sizes between classes. As the sample size increases, the two-stage approach provides comparable sensitivity with better specificity in variable selection. In binary classification and multicategory classification with balanced sample sizes, the two-stage approach provides comparable variable selection and prediction accuracy as the GLM version and is computationally more efficient.

R package, datasets and results of computational experiments on additional publicly available gene expression datasets are available in the online supplements.

Key Words: Partial least squares; Classification; Variable selection; Dimension reduction; Two-stage PLS; Iteratively re-weighted partial least squares; Gene expression.

Dongjun Chung is PhD student, Department of Statistics, University of Wisconsin, Madison, WI 53706 (E-mail: chungdon@stat.wisc.edu). Sündüz Keleş is Associate Professor, Department of Statistics and Department of Biostatistics and Medical Informatics, University of Wisconsin, Madison, WI 53706 (E-mail: keles@stat.wisc.edu).

1 Introduction

Partial least squares (PLS), a well known dimension reduction method (Wold 1966) in chemometrics, has been gaining a lot of attention in high dimensional classification problems of computational biology. Traditionally designed for continuous response, PLS has been promoted as a multivariate linear regression method that can deal with large number of predictors (p), small sample size (n), and high collinearity among predictors. PLS operates by forming linear combinations of the predictors in a supervised manner, i.e., using the response, and then regresses the response on these latent variables. It can handle both univariate and multivariate response and is computationally fast. Furthermore, projection of the whole data on a low dimensional space facilitates graphical representation of the data. All of these properties make PLS an attractive candidate for high dimensional genomic data problems such as classification of tumor samples that are in the order of tens or hundreds based on thousands of features, i.e., gene expression.

Since PLS is originally designed for continuous response, its adaption to classification for high dimensional data is relatively recent. Barker and Rayens (2003) justified use of PLS for classification by establishing its connection to Fisher's linear discriminant analysis. Previous work that utilizes PLS for high dimensional data classification can be grouped into two. Nguyen and Rocke (2002a), Nguyen and Rocke (2002b), and Boulesteix (2004) employ a two-stage procedure. We will refer to this procedure as PLSDA (PLS Discriminant Analysis) throughout the manuscript. In the first stage, response is treated as a continuous variable and PLS is employed to construct latent variables that are linear combinations of the original predictors. In the subsequent step, an *off-the-shelf* classification method is used since the number of latent variables constructed in the first stage is usually much smaller than the sample size. As a consequence, the large p, small n problem often diminishes. Logistic regression, linear discriminant analysis, and quadratic discriminant analysis are some of the examples of the classification methods used in the second step. When the response is multicategorical, the first step in this procedure is replaced by multivariate response PLS by transforming the original categorical response into a numerical response matrix using dummy coding. The second line of work for PLS classification (Marx 1996; Ding and Gentleman 2004; Fort and Lambert-Lacroix 2005) incorporates PLS into a Generalized Linear Model (GLM) framework, referred to as GPLS hereafter. In GLMs, the log likelihood is usually maximized using the Newton-Raphson algorithm which in turn results in the iteratively re-weighted least squares (IRLS) method. Marx (1996), Ding and Gentleman (2004), and Fort and Lambert-Lacroix (2005) adopt PLS for classification by solving the weighted least squares problem arising within the IRLS method with PLS. Marx (1996) proposed such an approach for general GLMs and Ding and Gentleman (2004) studied it specifically for the classification problems and developed the multinomial regression version. Ding and Gentleman (2004) also applied Firth's procedure (Firth 1993) in order to avoid the common non-convergence and infinite parameter estimates problems of the logistic regression in large p, small n problems. In contrast, Fort and Lambert-Lacroix (2005) incorporated a ridge penalty within the GLM framework for the same purpose.

Boulesteix (2004) studied classification with PLSDA in depth across many high dimensional cancer datasets and concluded that it performs competitively with the best state-of-the-art classification methods such as K-Nearest Neighbours, Support Vector Machines and PAM (Tibshirani et al. 2002) for such datasets. In the computational experiments of Ding and Gentleman (2004), GPLS achieved lower classification error rates than the two-stage PLSDA especially for the multicategory classification in high dimensional expression datasets.

Although PLS can deal with more predictors than there are samples, all of the above approaches often utilize variable filtering as a pre-processing step before the PLS fit. Selecting variables based on two-sample t-test statistic is commonly used for binary classification (Nguyen and Rocke 2002b). For multicategory classification, all pairwise t-filter proposed by Nguyen and Rocke (2002a) or ratio of the between-sum-of-squares to within-sum-of-squares (BSS/WSS) are commonly used (Boulesteix 2004). Once the variables are ranked based on a criterion, a subset of high ranking variables are further passed down to PLSDA or GPLS. Boulesteix (2004) established that ordering of the variables based on the BSS/WSS approach coincides with the ordering produced by the absolute values of the coefficients for the first PLS component. Although these pre-selection approaches often improve the performance of PLS classification by filtering out noise, their choice is often arbitrary and there is no established and computationally easy way of deciding what number of top ranking variables should be passed down to PLS classification. Furthermore, commonly used variable filtering approaches are all univariate and ignore correlations among variables. Recently, Chun and Keleş (in press) provided both theoretical and empirical results that the performance of PLS is ultimately affected by the large number of predictors in modern genomic data analysis. In particular, existence of high number of irrelevant variables leads to inconsistency of coefficient estimates in the linear regression setting. As a result, Chun and Keleş (in press) proposed sparse partial least squares (SPLS) regression which promotes variable selection within the course of PLS dimension reduction. Specifically, SPLS imposes sparsity when constructing the direction vectors, thereby the resulting latent variables depend only on a subset of the original set of predictors. This sparsity principle provides easy interpretation and correlations among the covariates are also well taken care of using the PLS framework. Moreover, SPLS is computationally efficient with a tunable sparsity parameter.

In this paper, we propose two new methods extending SPLS to classification problems. The first is SPLS discriminant analysis (SPLSDA) and the second is Sparse Generalized PLS (SGPLS). Both of these improve the two lines of PLS classification approaches reviewed above by employing variable selection and dimension reduction simultaneously. Our computational experiments indicate that SPLSDA and SGPLS outperform their PLS counterparts and in general perform comparably. We further establish that SGPLS has higher sensitivity than SPLSDA in variable selection when the classes are highly unbalanced in terms of their sample sizes. However, as the sample sizes increase, the variable selection performance of SPLSDA improves. It shows comparable variable selection sensitivity but has better specificity. The rest of the paper is organized as follows. The next section briefly reviews SPLS within the context of linear regression. In Section 3, we introduce SPLS-based classification approaches for both binary and multicategory responses. We present simulation studies and computational experiments with real datasets and compare our methods to competing ones in Sections 4 and 5. We end with a discussion in Section 6.

2 Simultaneous Dimension Reduction and Variable Selection with SPLS

Let $\boldsymbol{Y}_{n\times q}$ and $\boldsymbol{X}_{n\times p}$ represent the column centered response and the predictor matrices, respectively. PLS regression assumes latent components $\boldsymbol{T}_{n\times K}$ underlying both $\boldsymbol{Y}$ and $\mathbf{X}$. Hence, the PLS model is given by $\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{Q}^{\boldsymbol{T}} + \boldsymbol{F}$ and $\boldsymbol{X} = \boldsymbol{T}\boldsymbol{P}^{\boldsymbol{T}} + \boldsymbol{E}$, where $\boldsymbol{P}_{p\times K}$ and $\boldsymbol{Q}_{q\times K}$ are coefficients (loadings) and $\boldsymbol{E}_{n\times p}$ and $\boldsymbol{F}_{n\times q}$ are errors. The latent components $\boldsymbol{T}$

are defined as $\boldsymbol{T} = \boldsymbol{X}\boldsymbol{W}$, where $\boldsymbol{W}_{p \times K}$ are K direction vectors ($1 \leq K \leq \min\{n, p\}$). The main machinery of PLS is to find these direction vectors. The k-th direction vector $\hat{\boldsymbol{w}}_{\boldsymbol{k}}$ is obtained by solving the following optimization problem,

$$\max_{\boldsymbol{w}} \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{M} \boldsymbol{w} \quad \text{subject to} \quad \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{w} = 1 \quad \text{and} \quad \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{S}_{\boldsymbol{XX}} \hat{\boldsymbol{w}}_{\boldsymbol{l}} = 0 \quad l = 1, \cdots, k-1, \tag{1}$$

where $\boldsymbol{M} = \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{Y} \boldsymbol{Y}^{\boldsymbol{T}} \boldsymbol{X}$ and $\boldsymbol{S}_{\boldsymbol{XX}}$ represents the sample covariance matrix of the predictors. For univariate PLS, this objective function can be interpreted as follows (Frank and Friedman 1993):

$$\max_{\boldsymbol{w}} \operatorname{cor}^2 (\boldsymbol{Y}, \boldsymbol{X}\boldsymbol{w}) \operatorname{var} (\boldsymbol{X}\boldsymbol{w}) .$$

SPLS (Chun and Keleş in press) incorporates variable selection into PLS by solving the following minimization problem instead of the original PLS formulation (1):

$$\min_{\boldsymbol{w}, \boldsymbol{c}} -\kappa \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{M} \boldsymbol{w} + (1 - \kappa) \left(\boldsymbol{c} - \boldsymbol{w}\right)^{\boldsymbol{T}} \boldsymbol{M} \left(\boldsymbol{c} - \boldsymbol{w}\right) + \lambda_1 \left\|\boldsymbol{c}\right\|_1 + \lambda_2 \left\|\boldsymbol{c}\right\|_2 , \tag{2}$$

subject to $\boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{w} = 1$, where $\boldsymbol{M} = \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{Y} \boldsymbol{Y}^{\boldsymbol{T}} \boldsymbol{X}$. This formulation promotes exact zero property by imposing L_1 penalty onto a surrogate of direction vector ($\boldsymbol{c}$) instead of the original direction vector ($\boldsymbol{w}$), while keeping $\boldsymbol{w}$ and $\boldsymbol{c}$ close to each other. Here, L_2 penalty takes care of the potential singularity of $\boldsymbol{M}$. This formulation can be solved efficiently as described in Chun and Keleş (in press). One special case is worth mentioning here. If the response $\boldsymbol{Y}$ is univariate, then the solution of this formulation results in a soft thresholded direction vector:

$$\hat{\boldsymbol{c}} = \left(|\boldsymbol{Z}| - \lambda_1/2\right)_+ sign\left(\boldsymbol{Z}\right),$$

where $\boldsymbol{Z} = \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{Y} / ||\boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{Y}||$ and $(x)_+ = \max(0, x)$. Chun and Keleş (in press) recast this soft thresholding as

$$\hat{\boldsymbol{c}} = \left(|\boldsymbol{Z}| - \eta \max_{1 \leq j \leq p} |Z_j|\right)_+ sign\left(\boldsymbol{Z}\right),$$

where $0 \leq \eta \leq 1$ and justify setting $0 < \kappa \leq 0.5$ and $\lambda_2 = \infty$. Therefore, we have two key tuning parameters, η and K, in this formulation. Note that controlling η instead of the direction vector specific sparsity parameters λ_k, $k = 1, \cdots, K$, avoids combinatorial tuning of the set of sparsity parameters and provides a bounded range for the sparsity parameter, i.e., $0 \leq \eta \leq 1$. Furthermore, if $\eta = 0$, then SPLS coincides with PLS.

3 Classification with SPLS

3.1 SPLS Discriminant Analysis

SPLSDA (SPLS Discriminant Analysis) is a direct application of the original SPLS and closely follows the PLSDA approach of Nguyen and Rocke (2002a,b). We first construct latent components using SPLS regression by treating the categorical response as a continuous variable. For binary response, we use $\{0, 1\}$ dummy coding. We will provide below some further motivation for this choice. For multicategorical response, we utilize the *reference cell coding* employed by Nguyen and Rocke (2002a) and Boulesteix (2004). This coding scheme assumes that the response can be one of the $(G + 1)$ classes denoted by $0, 1, \cdots, G$, where 0 is the 'baseline' class, e.g., control group. Then, the recoded response matrix is defined as a $n \times (G + 1)$ matrix with elements $y^*_{i,(g+1)} = I(y_i = g)$ for $i = 1, \cdots, n$ and $g = 0, 1, \cdots, G$, where $I(A)$ is an indicator function of event A. The resulting response matrix $\boldsymbol{Y}^*$ is column centered before the SPLS fit. After construction of the latent components, we fit a classifier.

For the last step of SPLSDA, we can choose from multiple classification methods because the number of latent components (K) is usually much smaller than the sample size n. Linear classifiers such as linear discriminant analysis (LDA) and logistic regression are commonly used. In this context, linear classifiers might be more preferable from the interpretation point of view. Let $\hat{\boldsymbol{\beta}}^{LC}$ denote coefficient estimates of the latent components from a linear classifier. Then, we can obtain coefficient estimates for the original predictors as $\hat{\boldsymbol{\beta}} = \boldsymbol{W}\hat{\boldsymbol{\beta}}^{\boldsymbol{LC}}$ because $\boldsymbol{T}\hat{\boldsymbol{\beta}}^{LC} = \boldsymbol{X}\boldsymbol{W}\hat{\boldsymbol{\beta}}^{LC} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$.

A curious aspect of the above procedure is treating the categorical response as a continuous variable and the choice of the dummy coding. We next investigate this aspect for both multicategory and binary classification. Proofs of theorems 1 and 2 are provided in Appendix.

Theorem 1. *Consider the response matrix $\boldsymbol{Y}$ with reference cell coding and let $\boldsymbol{Y}^*$ represent its column centered version. Let $\hat{\mu}_{j,g}$ and n_g be the sample mean of the j-th predictor in class g and the sample size of class g, respectively. Let n be the overall sample size, i.e., $n = n_0 + n_1 + \cdots + n_G$, and n_{-g} be the sample size excluding g-th class, i.e., $n_{-g} = n - n_g$. Then, the first direction vector of SPLS is obtained by*

$$\hat{\boldsymbol{c}} = argmax_{\boldsymbol{c}} \sum_{g=0}^{G} \left(\frac{n_g n_{-g}}{n}\right)^2 \left(\sum_{j=1}^{p} c_j (\hat{\mu}_{j,g} - \hat{\mu}_{j,-g})\right)^2 - \lambda_1 \left\|\boldsymbol{c}\right\|_1,$$

where c_j is the j-th element of the direction vector $\boldsymbol{c}$.

This result indicates that contribution of each class to the construction of direction vectors is affected simultaneously by both the class sample size (through $(n_g n_{-g}/n)$) and the discrepancy between the within- and out-of-class sample means across predictors. As a result, the first direction vector of SPLS is likely to be most affected by the class with larger sample size when the effect sizes of the predictors across the classes are comparable. We next consider the binary classification case where the effect of class size diminishes since $(n_g n_{-g}/n)$ is the same for both classes.

Theorem 2. *Consider the binary response $\boldsymbol{Y}$ with $\{0, 1\}$ dummy coding. Then, the first direction vector of SPLS with the mean centered response $\boldsymbol{Y}^*$ has components of the form:*

$$\hat{c}_j = a \left(|\hat{\mu}_{j,1} - \hat{\mu}_{j,0}| - \lambda_1^*\right)_+ sign\left(\hat{\mu}_{j,1} - \hat{\mu}_{j,0}\right), \quad j = 1, \cdots, p,$$

where λ_1^* is a linear function of the sparsity parameter λ_1 on the first direction vector and a is some positive constant which does not depend on j. Moreover, if the columns of the predictor matrix are scaled to unit variance, then

$$\hat{c}_j = a \left(f(|t_j|) - \lambda_1^*\right)_+ sign\left(t_j\right), \quad j = 1, \cdots, p,$$

where t_j is the two sample t-statistic for j-th variable and f is some strictly monotone function that does not depend on j.

In summary, for binary classification, the j-th element of the first direction vector of SPLSDA is equivalent to the soft thresholded difference of the sample means of the two classes for the j-th predictor up to a scalar. Moreover, if the columns of the predictor matrix are scaled to unit variance, then it is equivalent to soft thresholded two sample t-statistic for the j-th predictor up to a scalar. This result is an extension of the property that ordering of the variables based on the first PLS component coincides with the ordering based on the widely used ratio of between to within sum-of-squares (Boulesteix 2004). Furthermore, it establishes the connection of SPLSDA to variable filtering with two sample t-statistics that is commonly employed for binary classification. As a result, SPLSDA has the ability to include variables that variable filtering would select in construction of very first direction vector. Moreover, it can select additional variables, i.e., variables that become significant once the response is adjusted for other variables, in the construction of the subsequent direction vectors.

This simple extension of SPLS for classification has two appealing properties. First, it inherits the simultaneous dimension reduction and variable selection property of SPLS. Second, it is computationally efficient since it only requires computational time of one run of SPLS and a classifier.

3.2 Sparse Generalized Partial Least Squares (SGPLS)

We next develop SGPLS (Sparse Generalized PLS) as a more principled version of SPLS classification. It extends SPLS to the GLM framework. Consider the logistic regression model with the logit link function:

$$\log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{x_i^T}\boldsymbol{\beta},$$

where $p_i = P(y_i = 1 \mid \boldsymbol{x_i})$ and $\boldsymbol{x_i}$ is the i-th row vector of $\boldsymbol{X}$. The log likelihood can be explicitly written as

$$l\left(\boldsymbol{\beta}\right) = \sum\nolimits_{i=1}^{n} \left\{ y_i \boldsymbol{x_i^T}\boldsymbol{\beta} - \log\left(1 + exp(\boldsymbol{x_i^T}\boldsymbol{\beta})\right)\right\}.$$

This minimization problem can be solved with the Newton-Raphson algorithm which results in the iteratively re-weighted least squares (IRLS). Specifically, for the current estimates $\tilde{\boldsymbol{\beta}}$, we solve the following weighted least squares problem:

$$\min_{\boldsymbol{\beta}} \sum_{i=1}^{n} v_i \left(z_i - \boldsymbol{x_i^T \beta} \right)^2, \tag{3}$$

where

$$\begin{aligned} \tilde{p}_i &= exp(\boldsymbol{x_i^T \tilde{\beta}}) / \left(1 + exp(\boldsymbol{x_i^T \tilde{\beta}}) \right) \quad \text{(estimated success probability)}, & (4) \\ z_i &= \boldsymbol{x_i^T \tilde{\beta}} + (y_i - \tilde{p}_i) / \left(\tilde{p}_i \left(1 - \tilde{p}_i \right) \right) \quad \text{(working response)}, & (5) \\ v_i &= \tilde{p}_i \left(1 - \tilde{p}_i \right) \quad \text{(weights)}. & (6) \end{aligned}$$

SPLS can be incorporated into the GLM framework by solving this weighted least squares problem using SPLS. In particular, the direction vectors of SGPLS are obtained by solving the following optimization problem:

$$\min_{\boldsymbol{w},\boldsymbol{c}} -\kappa \boldsymbol{w^T M w} + (1-\kappa) \boldsymbol{(c-w)^T M (c-w)} + \lambda_1 \left\| \boldsymbol{c} \right\|_1 + \lambda_2 \left\| \boldsymbol{c} \right\|_2 \tag{7}$$

subject to $\boldsymbol{w^T w} = 1$, where $\boldsymbol{M = X^T V z z^T V X}$, $\boldsymbol{V}$ is a diagonal matrix with entries v_i, and $\boldsymbol{z} = (z_1, \cdots, z_n)$ is the vector of working responses. When $\boldsymbol{w}$ and $\boldsymbol{c}$ become close to each other at convergence, the resulting direction vector can be interpreted as follows:

$$\hat{\boldsymbol{c}} = \arg\max_{\boldsymbol{c}} \left\{ \text{cor}^2 \left(\boldsymbol{V}^{1/2} \boldsymbol{z}, \boldsymbol{V}^{1/2} \boldsymbol{X c} \right) \text{var} \left(\boldsymbol{V}^{1/2} \boldsymbol{X c} \right) - \lambda_1 \left\| \boldsymbol{c} \right\|_1 - \lambda_2 \left\| \boldsymbol{c} \right\|_2 \right\}.$$

In the case of univariate response, the solution of the above formulation coincides with an univariate soft thresholded direction vector. Specifically, the solution is given by

$$\hat{\boldsymbol{c}} = (|\boldsymbol{Z}| - \eta \max_{1 \leq j \leq p} |Z_j|)_+ sign(\boldsymbol{Z}),$$

where $\boldsymbol{Z = X^T V z / || X^T V z ||}$.

SGPLS can be generalized to the multicategorical response by considering the following multinomial model:

$$\log \left(\frac{p_{ig}}{p_{i0}} \right) = \boldsymbol{x_i^T \beta_g},$$

for $g = 1, \cdots, G$ and $p_{ig} = P(y_i = g \mid \boldsymbol{x_i})$. We set the coefficient $\boldsymbol{\beta_0}$ for the baseline to zero for identifiability. In the multinomial logistic regression model, the log-likelihood to be maximized is given by:

$$l\left(\boldsymbol{\beta}\right) = \sum\nolimits_{i=1}^{n} \left\{ \sum\nolimits_{g=1}^{G} y_{ig} \boldsymbol{x_i^T} \boldsymbol{\beta_g} - \log \left(1 + \sum\nolimits_{g=1}^{G} exp(\boldsymbol{x_i^T} \boldsymbol{\beta_g}) \right) \right\}.$$

This log likelihood can be maximized using the Newton-Raphson algorithm as in the case of logistic regression model. However, since all the regression coefficients are replaced with their vector counterparts to accommodate multicategory response, such a naive implementation requires large sparse matrix calculations. In order to avoid this, we iteratively solve the profile log likelihood for $\boldsymbol{\beta_g}$ while fixing $\boldsymbol{\beta_l}, l \neq g$, for $g, l = 1, \cdots, G$. Specifically, for the current estimates $\tilde{\boldsymbol{\beta}}$ and class g, we solve the following weighted least squares problem:

$$\min_{\boldsymbol{\beta_g}} \sum\nolimits_{i=1}^{n} v_{ig} \left(z_{ig} - \boldsymbol{x_i^T} \boldsymbol{\beta_g} \right)^2, \tag{8}$$

where

$$\begin{aligned} \tilde{p}_{ig} &= exp(\boldsymbol{x_i^T} \tilde{\boldsymbol{\beta}}_{\boldsymbol{g}}) / \left(1 + \sum_{g=1}^{G} exp(\boldsymbol{x_i^T} \tilde{\boldsymbol{\beta}}_{\boldsymbol{g}}) \right) \quad \text{(estimated success probability)}, & (9) \\ z_{ig} &= \boldsymbol{x_i^T} \tilde{\boldsymbol{\beta}}_{\boldsymbol{g}} + \left(y_{ig} - \tilde{p}_{ig} \right) / \left(\tilde{p}_{ig} \left(1 - \tilde{p}_{ig} \right) \right) \quad \text{(working response)}, & (10) \\ v_{ig} &= \tilde{p}_{ig} \left(1 - \tilde{p}_{ig} \right) \quad \text{(weights)}. & (11) \end{aligned}$$

This problem has a similar form to that of the binary classification. Hence, we can obtain the weighted SPLS direction vectors similarly. As a result, we avoid large matrix calculations and fully take advantage of the block diagonality of the weight matrix. Our computational experiments indicated that this approach is computationally more efficient than the naive Newton-Raphson approach (data not shown). A similar approach was also employed in Friedman et al. (2008). SGPLS can be summarized as an algorithm as follows:

```
SGPLS algorithm (binary and multicategorical response)

  1. Initialize the number of iterations $nstep = 0$, active set, i.e. set of selected variables for SPLS A $= \{\}$, and change in estimates $\Delta\hat{\beta} = \infty$. Initialize estimated success probabilities as follows using the non-centered version $\boldsymbol{Y}^{**}$ of the dummy coded response: $\hat{p}_i = (y_i^{**} + 0.5)/2$ for binary response and $\hat{p}_{ig} = 2(y_{ig}^{**} + 0.5)/(G + 3)$ for multicategorical response.

  2. While ( $\Delta\hat{\beta} > \epsilon$ and $n.step < max.step$ ),

    (a) • Binary response.
            i. If $n.step > 0$, calculate $\hat{p}_i$ as in (4) for the current estimate $\hat{\boldsymbol{\beta}}^{current}$.
           ii. Update $z_i$ and $v_i$ given in (5) and (6) using $\hat{p}_i$.
          iii. Obtain $\hat{\boldsymbol{\beta}}$ by solving the SGPLS equation in (7) for given $\eta$ and $K$.
        • Multicategorical response.
          For $g = 1, \cdots, G$:
            i. If $n.step > 0$, calculate $\hat{p}_{ig}$ as in (9) for the current estimate $\hat{\boldsymbol{\beta}}^{current}$.
           ii. Update $z_{ig}$ and $v_{ig}$ given in (10) and (11) using $\hat{p}_{ig}$.
          iii. Obtain $\hat{\boldsymbol{\beta}}_{\boldsymbol{g}}$ by solving the SGPLS equation in (7) with given $\eta$ and $K$.
    (b) Update $\Delta\hat{\beta} = mean(|\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}^{current}|)/mean(|\hat{\boldsymbol{\beta}}^{current}|)$, $\hat{\boldsymbol{\beta}}^{current} = \hat{\boldsymbol{\beta}}$, and $nstep = nstep + 1$.
```

4 Simulation Studies

Boulesteix (2004) compared PLS-based dimension reduction and classification with some of the best stat-of-the-art classification methods on several datasets with computational experiments and illustrated that it is best in terms of classification accuracy for most of the datasets. Therefore, in this section, we compare and contrast our proposed methods, SPLSDA and SGPLS, with PLSDA and GPLS which utilize PLS principle without built-in variable selection. Previous work for both PLSDA and GPLS show that their classification accuracy might depend on the number of variables included as a result of variable filtering a priori to PLS fit. Hence, we also consider versions of PLSDA and GPLS with variable filtering. Specifically, we consider t-statistic for binary classification and all pairwise t-filter for multicategory classification. For binary classification, the two sample t-statistic is calculated for all the variables. Then, variables are ranked according to their absolute value of t-statistics and the *top* ranking variables are passed down to PLS fit. There are no well-established rules for thresholding the ranked list of variables (Boulesteix 2004). In our simulations, we treat the top m variables to include as a tuning parameter. For multicategory classification, all $(G+1)G/2$ pairwise absolute mean differences of j-th variable, $|\hat{\mu}_{j,g} - \hat{\mu}_{j,g\prime}|$, are computed and then compared to a critical score,

$$t_{\alpha/2,n-(G+1)}\sqrt{MSE\left(\frac{1}{n_g}+\frac{1}{n_{g\prime}}\right)},$$

for $g < g\prime \in \{0, 1, \cdots, G\}$, where MSE is the estimate of variability from the analysis of variance (ANOVA) model with one factor and $(G + 1)$ groups. Then, PLS is fit using the variables for which m of the pairwise absolute mean differences exceed the critical score. Here, m is treated as a tuning parameter.

4.1 Simulations for Binary Classification

We set the data generating mechanism for binary classification as follows by first generating three latent variables H_1, H_2, and H_3 from $\mathcal{N}(0, 5^2)$. Then, covariates were generated as $x_1, \cdots, x_5 = H_1 + \mathcal{N}(0, 1)$, $x_6, \cdots, x_{10} = H_2 + \mathcal{N}(0, 1)$, $x_{11}, \cdots, x_{15} = H_3 + \mathcal{N}(0, 1)$, and $x_{16}, \cdots, x_p \sim \mathcal{N}(0, 1)$. The matrix of covariates was scaled to have mean zero and variance one. In order to generate response, we first generated $p = P(Y = 1 \mid H_1, H_2, H_3) = g(3H_1 - 4H_2)$, where g is the inverse of the link function. Then, the binary response was generated as $Y \sim Bernoulli(p)$. In this set-up, we observe groups of surrogates, $x_1 - x_5$ and $x_6 - x_{10}$, of the true covariates that affect the response.

In our comparisons, logistic regression was used as the classifier in SPLSDA and PLSDA since GPLS and SGPLS directly utilize logistic regression. For all of these methods, number of components K was searched over $1, 2, \cdots, 5$ and η was searched over $0.1, 0.2, \cdots, 0.9$ in SPLSDA and SGPLS. For variable filtering in PLSDA and GPLS, we searched m over both 10%-100% with increments of 10% and 1%-100% with increments of 1%.

Table 1: Variable selection performance and classification accuracy of PLSDA, GPLS, SPLSDA, and SGPLS for binary classification. For PLSDA and GPLS, two versions of variable filtering are considered. Variable filtering tuning for PLSDA-g1 and GPLS-g1 is over $10\%, 20\%, \cdots, 100\%$, whereas for PLSDA-g2 and GPLS-g2, a finer grid of $1\%, 2\%, \cdots, 100\%$ is used. Misclassification: number of misclassified observations among 1000 test observations. A: number of variables selected among 1000. H: number of true variables in A. The maximum value of H is 10. Reported are the median values across 100 simulation runs. Numbers in parentheses are the corresponding interquantile ranges (IQRs). For PLSDA and GPLS, A and H are always 1000 and 10, respectively, by construction.

Method	K	η	Misclassification (1000)		A (1000)	H (10)
PLSDA	1	-	248	(35.25)	1000 (0)	10 (0)
GPLS	1	-	246.5	(35.25)	1000 (0)	10 (0)
PLSDA-g1	1	-	169	(36.25)	100 (0)	10 (0)
GPLS-g1	1	-	172	(44)	100 (0)	10 (0)
PLSDA-g2	2	-	51	(40.5)	10 (0)	10 (0)
GPLS-g2	2	-	52	(30.75)	10 (0)	10 (0)
SPLSDA	1	0.8	68.5	(79.5)	10 (1)	10 (1)
SGPLS	2	0.9	58	(42.75)	10 (0)	10 (0)

We simulated 100 training and test datasets and tuned all the methods by 5-fold cross-validation for each dataset. We set $n = 100$ and $p = 1000$ for the training dataset and $n = 1000$ for the test dataset. Table 1 displays results over 100 simulation runs. These results indicate that SPLSDA and SGPLS perform similar to each other and outperform their counterparts PLSDA and GPLS. Consistent with previous results in literature, variable filtering improved the classification accuracy of PLSDA and GPLS and helped to capture the true set of variables. When the variables were searched over top $10\%, 20\%, \cdots, 100\%$, SPLSDA and SGPLS still significantly outperformed PLSDA and GPLS (PLSDA-g1 and GPLS-g1) both in classification accuracy and variable selection. When the variables were searched over top $1\%, 2\%, \cdots, 100\%$, PLSDA and GPLS (PLSDA-g2 and GPLS-g2) slightly outperformed SPLSDA and SGPLS. This is not unexpected because the variable filter is allowed to pick exactly the top 1% of the variables which is likely to coincide with the true set of variables. In this sense, the classification accuracy attained by PLSDA-g2 and GPLS-g2 might be close to the best possible accuracy for this simulation setting. However, searching over the finer grid for m in PLSDA-g2 and GPLS-g2 increased the computational time ten fold compared to PLSDA-g1 and GPLS-g1. In contrast, by just searching over ten values of the tuning parameter η ($\eta = 0.1, 0.2, \cdots, 0.9$), SPLSDA and SGPLS attained comparable classification accuracy to PLSDA-g2 and GPLS-g2 and selected only the true set of variables most of the time.

In order to understand the variable selection properties of SPLSDA and SGPLS, we examined the resulting coefficient estimates. Table 2 displays coefficient estimates for PLSDA, SPLSDA, GPLS, and SGPLS across three sets of covariates for 100 runs. For PLSDA and SPLSDA, the coefficient estimates were calculated as $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^{LC}$, where $\hat{\boldsymbol{W}}$ denotes the matrix of estimated direction vectors and $\hat{\boldsymbol{\beta}}^{\boldsymbol{LC}}$ is the vector of coefficient estimates obtained from the logistic regression. By construction, variables $x_1 - x_{10}$ correspond to true signals, i.e., relevant variables, and should have nonzero estimates. More specifically, variables $x_1 - x_5$ have positive relationships with the response while variables $x_6 - x_{10}$ have negative relationships. In contrast, variables $x_{11} - x_{1000}$ correspond to noise variables and should have zero estimates. On average, all of the four methods display this trend. However, for PLSDA and GPLS, the separation between the relevant and noise variables is much smaller compared to that of SPLSDA and SGPLS. Although PLSDA and GPLS have relatively small nonzero coefficients for the noise variables, the fact that these are not exactly zero increases misclassification rates about four fold compared to SPLSDA and SGPLS.

Table 2: Coefficient estimates of PLSDA, SPLSDA, GPLS, and SGPLS for 100 runs of the binary classification simulation. For PLSDA and SPLSDA, the coefficient estimates were calculated as $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^{LC}$, where $\hat{\boldsymbol{W}}$ is the matrix of estimated direction vectors and $\hat{\boldsymbol{\beta}}^{LC}$ is the vector of coefficient estimates of the latent components obtained from the logistic regression. First ten variables, corresponding to $\hat{\beta}_1 - \hat{\beta}_{10}$, represent true signals, by construction. "True" coefficients were estimated using only the true signals and details of the estimation procedure are provided in the text. Reported are the median values across 100 simulation runs. Numbers in the parentheses are the corresponding IQRs.

PLSDA/SPLSDA

Method	$\hat{\beta}_1 - \hat{\beta}_5$		$\hat{\beta}_6 - \hat{\beta}_{10}$		$\hat{\beta}_{11} - \hat{\beta}_{1000}$	
PLSDA	1.586	(1.160)	-2.166	(1.645)	-0.001	(0.447)
SPLSDA	1.962	(2.135)	-2.437	(2.932)	0	(0)

GPLS/SGPLS

Method	$\hat{\beta}_1 - \hat{\beta}_5$		$\hat{\beta}_6 - \hat{\beta}_{10}$		$\hat{\beta}_{11} - \hat{\beta}_{1000}$	
GPLS	0.078	(0.049)	-0.100	(0.070)	-0.000	(0.028)
SGPLS	0.291	(0.076)	-0.386	(0.061)	0	(0)
"true"	0.377	-	-0.501	-	0	-

Chun and Keleş (in press) illustrated that, in the liner regression context, PLS coefficient estimates are attenuated when the number of noise variables is overwhelmingly large. We investigated whether the same phenomena occurs within the context of PLS-based classification with GPLS. We first characterized the true values of the coefficients corresponding to the relevant variables when we are given only these variables and do not have the variable selection problem. Note that since the data generating process is not based on the covariates $x_1, \cdots, x_{10}$, "true" values of these coefficients are unknown per se. We generated $n = 20,000$ observations from the same data generating distribution and fit a logistic regression using only $x_1, \cdots, x_{10}$. We repeated this process 500 times and took the resulting means of the estimates as the true values. The estimated "true" coefficients are given in Table 2. Table 3 lists the ratios of squared bias, variance, and mean squared error of GPLS to SGPLS averaged within the three covariate groups. GPLS estimates were overall smaller in magnitude than the corresponding true values, confirming the attenuation effect. Furthermore, although imposing sparsity increased variability of the estimates of the relevant covariates compared to GPLS, the reduction in bias compensated this, leading to a smaller MSE for SGPLS. For the irrelevant variables, SGPLS outperformed GPLS both in bias and variance.

Table 3: Squared bias, variance and mean squared error (mse) ratios of GPLS to SGPLS across three sets of variables for binary classification.

	GPLS/SGPLS of		
Variables	Bias^2	Var	mse
$x_1 - x_5$	7.76	0.33	3.36
$x_6 - x_{10}$	11.44	0.36	4.40
$x_{11} - x_{1000}$	8.59	7.48	7.48

4.2 Simulations for Multicategory Classification

We considered a response with four classes in this simulation. All classes were set to have the same number of observations: $n_0 = n_1 = n_2 = n_3$, where $n_0 + n_1 + n_2 + n_3 = n$. The latent structure was generated as a matrix $H = [H_1, H_2, H_3]$, with rows corresponding to g-th class generated from $\mathcal{N}_3(\boldsymbol{\mu}_g, \boldsymbol{I})$, for $g = 0, 1, 2, 3$, where $\boldsymbol{\mu}_0 = (0, 0, 0)^T, \boldsymbol{\mu}_1 = (4, 0, 0)^T, \boldsymbol{\mu}_2 = (0, 4, 0)^T$, and $\boldsymbol{\mu}_3 = (0, 0, 4)^T$. Finally, we generated covariates $x_1, \cdots, x_5 = H_1 + 0.5\mathcal{N}(0, 1)$, $x_6, \cdots, x_{10} = H_2 + 0.5\mathcal{N}(0, 1)$, $x_{11}, \cdots, x_{15} = H_3 + 0.5\mathcal{N}(0, 1)$, and $x_{16}, \cdots, x_p = 0.5\mathcal{N}(0, 1)$. In this set-up, $x_1, \cdots, x_5$ discriminate between classes 0 and 1, $x_6, \cdots, x_{10}$ between classes

0 and 2, and $x_{11}, \cdots, x_{15}$ between classes 0 and 3. The covariate matrix was scaled to have mean zero and variance one. We performed 100 simulation runs and chose all the tuning parameters by 5-fold cross-validation for each dataset. We set $n = 100$ and $p = 1000$ for the training dataset and $n = 1000$ for the test dataset. GPLS is excluded from this comparison since it is computationally too slow to perform multiple simulation runs in this high dimensional setting.

Table 4 summarizes the results. As in the binary classification case, SPLSDA and SGPLS outperform PLSDA both in variable selection and classification accuracy. Although variable filtering improved the classification accuracy of PLSDA, SPLSDA and SGPLS still outperform PLSDA in terms of classification accuracy and capturing the true variables. SPLSDA performs only slightly worse than SGPLS in terms of classification accuracy; however, a closer look on the results reveals some differences between the two approaches.

Table 4: Variable selection performance and classification accuracy of PLSDA (PLSDA-g for PLSDA with variable filtering), SPLSDA, and SGPLS for multicategory classification. Misclassification: number of misclassified observations among 1000 test observations. A: number of variables selected among 1000. H: number of true variables in A. The maximum value of H is 15. Reported are the median values across 100 simulation runs. Numbers in the parentheses are the corresponding IQRs. For PLSDA, A and H are always 1000 and 15, respectively, by construction.

Method	K	η	Misclassification (1000)		A (1000)	H (15)
PLSDA	3	-	280	(37)	1000 (0)	15 (0)
PLSDA-g	3	-	104	(30.25)	36 (7)	15 (0)
SPLSDA	3	0.9	53	(22.25)	15 (1)	15 (0)
SGPLS	1	0.9	47.5	(10)	15 (0)	15 (0)

Table 5 displays the final coefficient estimates from SPLSDA ($\eta = 0.9$ and $K = 3$) and SGPLS ($\eta = 0.9$ and $K = 1$) for one of the simulated datasets. For SPLSDA, estimates were calculated as in the binary classification simulation. Recall that for the multicategorical response, we have a 1000 dimensional vector of coefficients for each class compared to the reference class. Both of the methods identify the correct set of variables, i.e., variables 1 to 15 in this set-up. The sizes of the estimates reveal an interesting difference between SPLSDA and SGPLS. In SGPLS, variables that discriminate a given class from the baseline have non-zero estimates whereas the other variables have exactly zero estimates in the corresponding class coefficient vector. In contrast, for SPLSDA, variables that discriminate a given class from the baseline have high coefficient estimates (in absolute value), however, the rest of the relevant variables have non-zero, albeit relatively small, estimates in the corresponding class coefficient vector. This observation illustrates that, compared to SPLSDA, SGPLS might provide more insights into variables that discriminate different classes from the baseline. A good way to visualize such information is to look at the density of the latent components across different classes. Panels (a) to (c) in Figure 1 display the distribution of the first latent components (one for each class by construction) across different classes for SGPLS. As implied by Table 5, the first latent component for each class versus the baseline discriminates the corresponding class from the baseline. Panel (d) in Figure 1 displays the scatter plot of classes on the first and second SPLSDA latent components. This plot supports that SPLSDA finds the set of direction vectors discriminating all classes simultaneously. Therefore, it does not particularly highlight which variables are most important for each class.

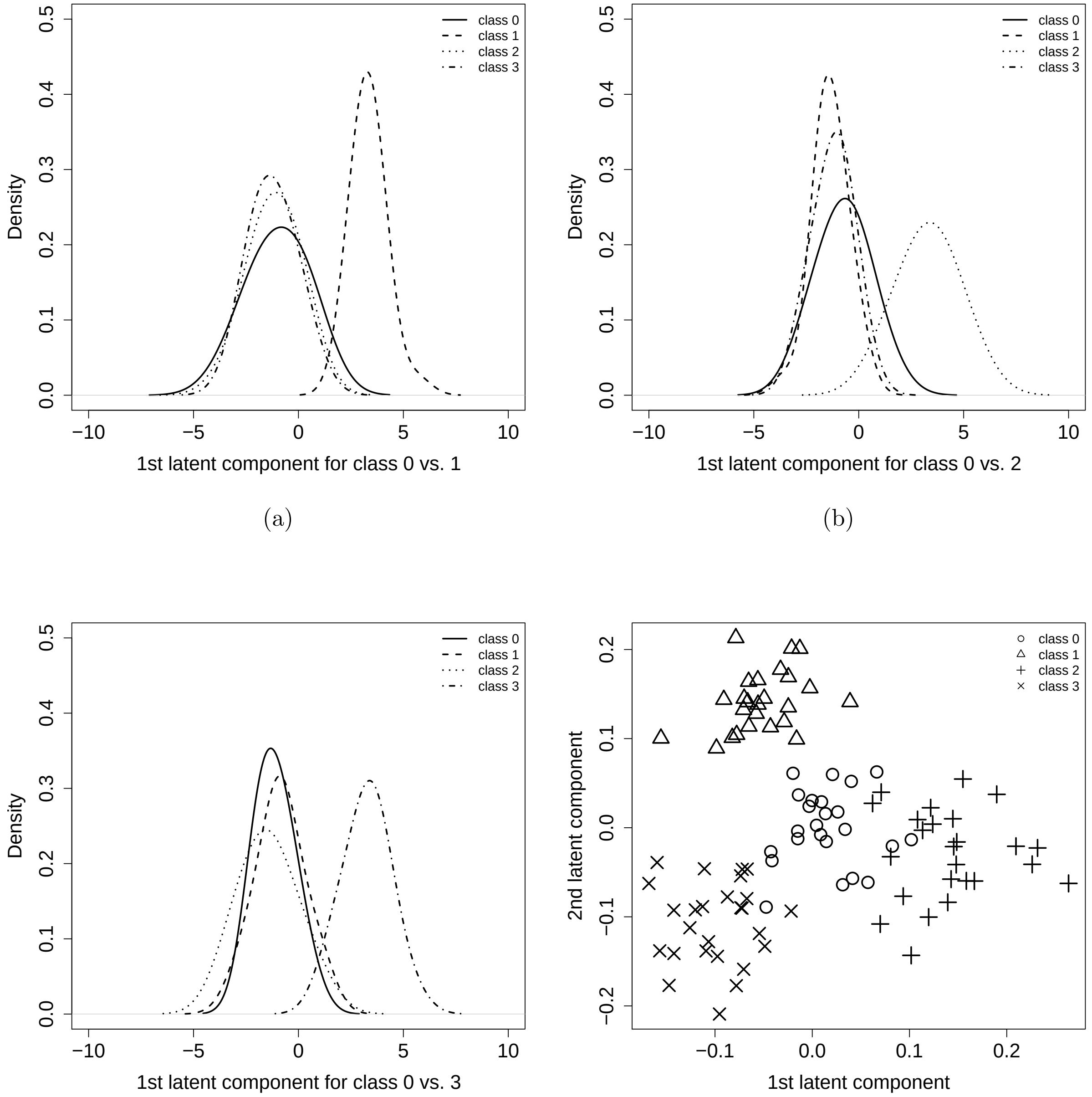
Table 5: Coefficient estimates of SPLSDA ($\eta = 0.9$ and $K = 3$) and SGPLS ($\eta = 0.9$ and $K = 1$) for one simulated dataset from the multicategory classification simulations. For SPLSDA, as in the binary classification simulation, the coefficient estimates were calculated as $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{W}}\hat{\boldsymbol{\beta}}^{LC}$, where $\hat{\boldsymbol{W}}$ are estimated direction vectors and $\hat{\boldsymbol{\beta}}^{LC}$ is the vector of coefficient estimates of the latent components obtained from the logistic regression. In each cell, the minimum and maximum values of the corresponding coefficient estimates list. Empty cells have zero coefficient estimates.

SPLSDA

	class 0 vs. 1	class 0 vs. 2	class 0 vs. 3
$\hat{\beta}_1 - \hat{\beta}_5$	(2.87, 2.96)	(1.11, 1.44)	(-0.29, -0.15)
$\hat{\beta}_6 - \hat{\beta}_{10}$	(-0.52, -0.24)	(3.35, 4.06)	(-0.29, 0.26)
$\hat{\beta}_{11} - \hat{\beta}_{15}$	(-0.29, -0.11)	(0.19, 0.54)	(3.44, 3.59)
$\hat{\beta}_{16} - \hat{\beta}_{1000}$			

SGPLS

	class 0 vs. 1	class 0 vs. 2	class 0 vs. 3
$\hat{\beta}_1 - \hat{\beta}_5$	(0.39, 0.40)		
$\hat{\beta}_6 - \hat{\beta}_{10}$		(0.42, 0.47)	
$\hat{\beta}_{11} - \hat{\beta}_{15}$			(0.47, 0.49)
$\hat{\beta}_{16} - \hat{\beta}_{1000}$			



(a)

(b)

(c)

(d)

Figure 1: Panels (a) - (c) display the distribution of the first SGPLS latent components for each of class 1 versus class 0 (baseline), class 2 versus class 0, and class 3 versus class 0, respectively. Panel (d) is the scatter plot of classes on the first and second SPLSDA latent components.

4.3 Simulations for comparing SPLSDA and SGPLS in multicategory classification with unbalanced classes

In the above two simulation studies, SPLSDA and SGPLS exhibit more or less similar performances both in terms of classification accuracy and variable selection despite their structural differences. In this section, we perform a simulation study for investigating the implications of Theorem 1, that is, SPLSDA might be prone to missing variables that exclusively discriminate classes with small sample sizes under the assumption that the effect sizes, i.e., coefficients of the class specific variables, are comparable across classes. In contrast, since SGPLS finds direction vectors separately for each pair of g-th group versus the baseline (0-th class), it might be able to detect such variables. This is an important practical issue because such unbalanced class compositions often arise in real datasets. We considered the following multicategory simulation setting. We generated a response with three classes where one of the classes had much smaller sample size compared to others: $n_0 = n_1 = 4.5n_2$ and $n_2 = 10$. The latent structure was generated as a matrix $H = [H_1, H_2]$, with rows corresponding to the g-th class generated from $\mathcal{N}_2(\boldsymbol{\mu}_g, \boldsymbol{I})$ for $g = 0, 1, 2$, where $\boldsymbol{\mu}_0 = (0, 0)^T$, $\boldsymbol{\mu}_1 = (4, 0)^T$, and $\boldsymbol{\mu}_2 = (0, 4)^T$. Finally, we generated covariates $x_1, \cdots, x_5 = H_1 + 0.5\mathcal{N}(0, 1)$, $x_6, \cdots, x_{10} = H_2 + 0.5\mathcal{N}(0, 1)$, and $x_{11}, \cdots, x_p = 0.5\mathcal{N}(0, 1)$, with $p = 1000$. In this setup, $x_1, \cdots, x_5$ discriminate between classes 0 and 1, and $x_6, \cdots, x_{10}$ between classes 0 and 2. As in the previous section, we performed 100 simulation runs and chose all the tuning parameters by 5-fold cross-validation for each dataset.

Table 6 shows the results. When $n = 100$, SPLSDA missed the variables discriminating class 2 from class 0. In contrast, SGPLS picked up all of the 10 relevant variables at the expense of a larger active set compared to SPLSDA. Specifically, SGPLS selected some false positives for the class 0 vs. 1 comparison while it generally selected only the true signals for the class 0 vs. 2 comparison (the median of A is 18 for the class 0 vs. 1 comparison and 11 for the class 0 vs. 2 comparison). In other words, SGPLS sacrificed some specificity for the class 0 vs. 1 comparison in order to obtain better sensitivity for the class 0 vs. 2 comparison. The main reason for this is that SGPLS uses a common η to control variable selection for both of the class 0 vs. 1 and the class 0 vs. 2 comparisons. If we employ separate tuning parameters for each of the class comparisons, the specificity of SGPLS might improve at the expense of a significant increase in computational time required for tuning. Next, we investigated how the variable selection properties of SPLSDA and SGPLS change for the unbalanced case as the sample size increases. As Table 6 indicates, when $n = 300$, SPLSDA is able to choose all of the 10 relevant variables. This is consistent with Theorem 1 since larger sample size leads to better estimates for the discrepancy of the within- and out-of-class sample means. SGPLS has more false positives for the class 0 vs. 1 comparison and the specificity of SGPLS is not improved by larger sample size (the median of A is 65.5 for the class 0 vs. 1 comparison and 10.5 for the class 0 vs. 2 comparison).

Table 6: Variable selection performance and classification accuracy of SPLSDA and SGPLS for unbalanced multicategory classification. Two different sample sizes for the training dataset were considered ($n = 100$ and $n = 300$), while the sample sizes of classes have the same ratio ($n_0 : n_1 : n_2 = 9 : 9 : 2$). Misclassification: number of misclassified observations among 1000 test observations. A: number of variables selected among 1000. H: number of true variables in A. The maximum value of H is 10. Reported are the median values across 100 simulation runs. Numbers in the parentheses are the corresponding IQRs.

$n = 100; n_0 = n_1 = 45, n_2 = 10$

Method	K	η	Misclassification (1000)	A (1000)	H (10)
SPLSDA	2	0.9	124 (42)	6 (8)	5 (5)
SGPLS	2	0.7	84 (24.75)	27 (34)	10 (0)

$n = 300; n_0 = n_1 = 135, n_2 = 30$

Method	K	η	Misclassification (1000)	A (1000)	H (10)
SPLSDA	3	0.9	42 (21.25)	13 (21.5)	10 (0)
SGPLS	2	0.5	83 (21.25)	74 (141.25)	10 (0)

5 Computational Experiments for Tumor Classification with Microarray Gene Expression Data

In order to evaluate our proposed methods for classification with high dimensional microarray data, we considered various publicly available gene expression datasets. Here, we present detailed results for prostate (Singh et al. 2002) and lymphoma (Alizadeh et al. 2000) datasets which involve binary and multicategorical classification, respectively. Results on five additional publicly available datasets are provided as online supplements. Both of the datasets were pre-processed (arrays were normalized, imputed, log transformed, and standardized to zero mean and unit variance across genes) as described in Dettling (2004) and Dettling and Bühlmann (2002). The prostate dataset consists of 52 prostate tumor and 50 normal samples. As a result of pre-processing, we have expression of $p = 6,033$ genes across these $n = 102$ samples. The lymphoma dataset provides expression for $p = 4,026$ genes across $n = 62$ patients with 42 samples of diffuse large B-cell lymphoma (DLBCL), 9 samples of follicular lymphoma (FL), and 11 samples of chronic lymphocytic leukemia (CLL).

We performed the following computational experiment on these datasets. We fitted the classification methods using two thirds of the datasets as the training data, while the remaining one thirds were predicted as the test data This procedure was repeated 100 times and each method was tuned by 5-fold cross-validation using the training data from each of these partitions. For prostate dataset, the t-statistic variable filtering method was used by searching m over top $10, 20, 30, 40, 50, 100, 200, 500, 1000$, and 1500 variables using cross-validation. For lymphoma dataset, all pairwise t-filter was applied. In the lymphoma dataset, DLBCL group, which is the largest class, was used as the baseline.

Results for prostate data are presented in Table 7. All of the methods basically show similar classification accuracy by misclassifying about 3 test subjects out of 35. We note that variable filtering did not improve the classification accuracy of PLSDA or GPLS and this is inline with the observation of Boulesteix (2004) that, for PLS, variable selection is often unnecessary to achieve an excellent classification accuracy for datasets with many relevant variables and only a few thousand variables. Figure 2 provides a closer look on the variable selection properties of SPLSDA and SGPLS for a typical partition of the dataset. For this particular partition, SPLSDA selected 44 genes, 25 of which are among the 157 genes selected by SGPLS. Panels (a) and (b) of Figure 2 display heatmaps of the expression data for the selected genes of SPLSDA and SGPLS, respectively. Top 52 rows are tumor samples, denoted by "+", and bottom 50 rows are normal samples, denoted by "-". Horizontal side bar on top of each plot reflects the signs and magnitudes of the coefficient estimates. Columns (genes) are clustered according to their expression across the 102 samples. Genes unique to each method are indicated by "$*$" on the corresponding heatmaps. We observe that, for both of the methods, selected genes cluster nicely. This eludes to the group selection property of SPLS, that is, genes highly correlated with each other and the response are selected simultaneously.

Table 7 further displays the results for the lymphoma dataset. This dataset is relatively easy and all methods have less than 2 misclassifications with SGPLS showing the best classification accuracy. Here, variable filtering did not improve the classification accuracy

Table 7: Variable selection and classification accuracy of PLSDA, GPLS, SPLSDA, and SGPLS for the tumor classification with microarray gene expression data. PLSDA-g and GPLS-g refer to PLSDA and GPLS with variable filtering. Misclassification: number of misclassified observations among test observations. A: number of variables selected. Reported numbers are median values of 100 runs. Numbers in the parentheses are the corresponding IQRs.

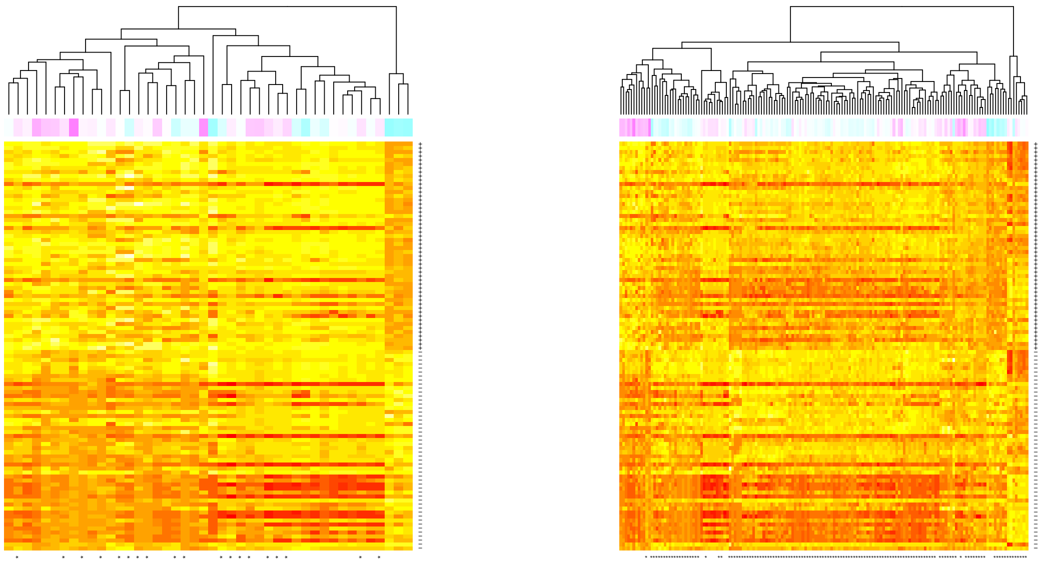
Binary response prostate dataset

Method	K	η	Misclassification (35)	A (6033)
PLSDA	4	-	3 (2)	6033 (0)
GPLS	4	-	3 (1)	6033 (0)
PLSDA-g	3	-	4 (3)	100 (482.5)
GPLS-g	3	-	3 (2)	100 (480)
SPLSDA	3	0.8	3 (2)	62.5 (242.75)
SGPLS	3	0.55	3 (2)	163 (836.5)

Multicategory response lymphoma dataset

Method	K	η	Misclassification (21)	A (4026)
PLSDA	2	-	1 (2)	4026 (0)
PLSDA-g	2	-	1 (2)	197 (1496)
SPLSDA	2	0.7	2 (2)	24.5 (38)
SGPLS	2	0.6	0 (1)	69 (133)

of PLSDA either. Figure 3 provides a closer look on the variable selection properties of SPLSDA and SGPLS for a typical partition of the dataset. Panel (a) displays the venn diagram comparison of the variables selected by each method. Note that, for SGPLS, we have an active set for each of DBLCL vs. FL and DBLCL vs. CLL classes. Panels (b)-(d) display heatmaps of the expression data for the selected genes for each method. In these heatmaps, top 11 rows are CLL samples, denoted by "+", middle 9 rows are FL samples, denoted by "=", and bottom 42 rows are DBLCL samples, denoted by "-". The horizontal side bars indicate the signs and magnitudes of the coefficient estimates. We observe that these coefficient estimates cluster well with respect to the expression values. In Panel (b), which displays the heatmap for SPLSDA, the expression values of the DBLCL group are clearly different from those of the FL and CLL groups. However, there is almost no difference in the expression values between FL and CLL samples. This indicates that SPLSDA mostly selected variables discriminating DBLCL group from the other two groups but it generally missed variables discriminating FL and CLL groups. In contrast, Panels (c) and (d) illustrate that SGPLS captured genes discriminating each of the FL and CLL classes from the DBLCL class separately in addition to genes discriminating both of the FL and



(a) SPLSDA

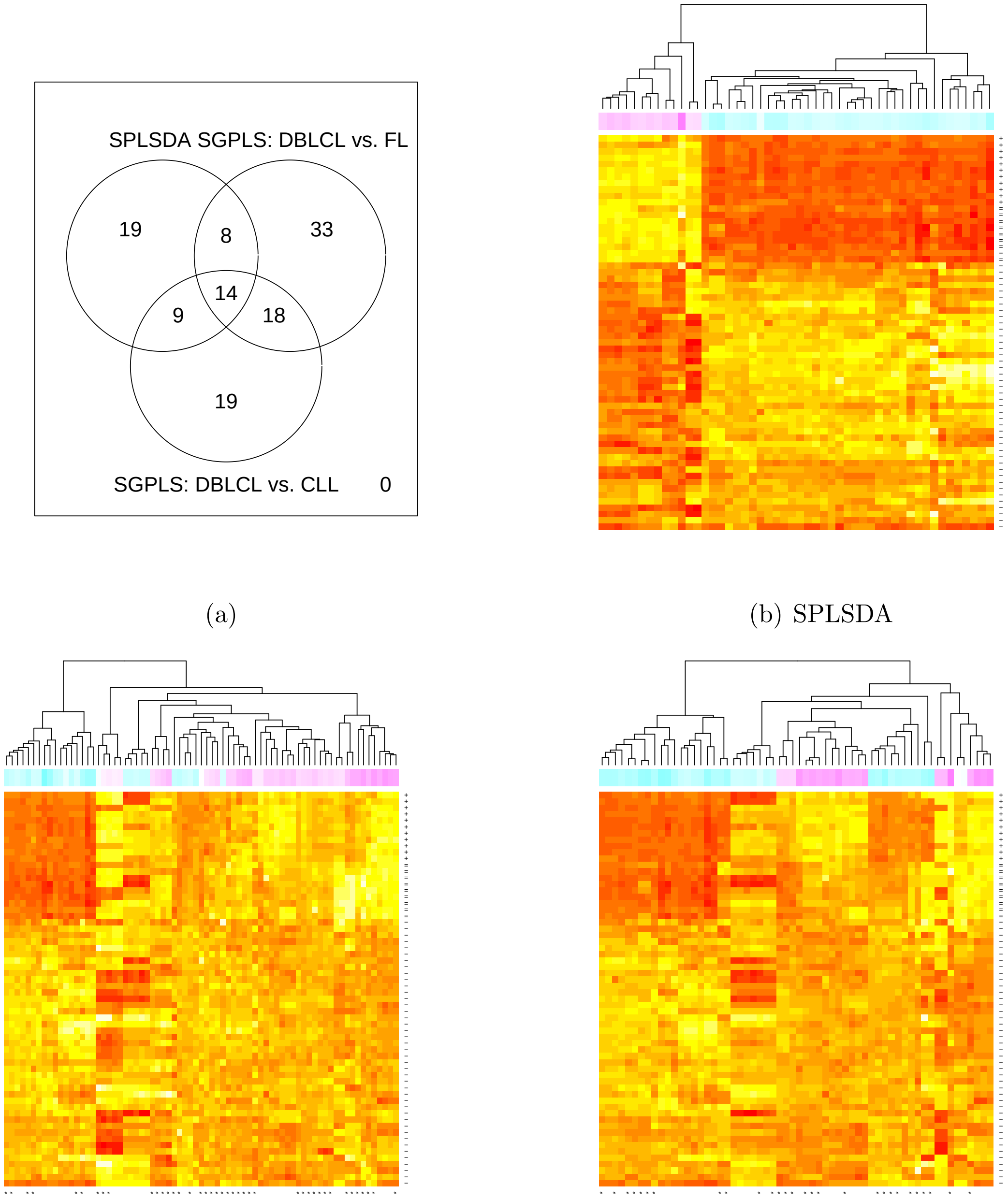
(b) SGPLS

Figure 2: Comparison of the variables selected by each method for the prostate dataset. (a) Heatmap of the expression values of the genes selected by SPLSDA. Genes specific to SPLSDA are marked by “$*$”. (b) Heatmap of the expression values of the genes selected by SGPLS. Genes specific to SGPLS are marked by “$*$”. Genes are clustered according to their expression across 50 normal (denoted by “-”) and 52 tumor (denoted by “+”) samples. Pink/blue colored horizontal side bar displays the coefficient estimates (pink for positive, blue for negative).

CLL classes from the DBLCL class.

6 Discussion

We proposed two approaches, SPLSDA and SGPLS, for adapting sparse partial least squares for classification. These approaches are natural extensions of prior work on classification of high dimensional data with PLS (Nguyen and Rocke 2002b,a; Boulesteix 2004; Ding and Gentleman 2004) to incorporate simultaneous variable selection and dimension reduction. We have observed that both of these approaches outperform their counterparts PLSDA and GPLS which lack built-in variable selection. By directly incorporating SPLS into a generalized linear model framework, SGPLS provides more insightful results especially for multicategory classification. Furthermore, it displays higher sensitivity in variable selection in



(a)

(b) SPLSDA

(c) SGPLS (DBLCL vs. FL)

(d) SGPLS (DBLCL vs. CLL)

Figure 3: Comparison of the variables selected by each method for the lymphoma dataset. (a) Venn diagram comparison of the selected genes with SPLSDA ($\eta = 0.7$ and $K = 2$) and SGPLS ($\eta = 0.6$ and $K = 2$). (b)-(d) Heatmaps of the expression values of the genes selected by each method. Genes are clustered according to their expression across 42 DBLCL (denoted by "-"), 9 FL (denoted by "=") and 11 CLL (denoted by "+") samples. The horizontal side bar displays the coefficient estimates (pink for positive, blue for negative). For SGPLS, genes specific to DBLCL vs. FL and to DBLCL vs. CLL comparisons are denoted by "*" in their corresponding heatmaps.

the case of multicategory classification with small unbalanced class sizes. Although SGPLS is a more principled approach that incorporates SPLS into generalized linear model framework, the two-stage approach SPLSDA which treats categorical response as numerical for dimension reduction performs only slightly worse in our binary class simulations, multicategory simulations with balanced class sizes, and majority of the computational experiments with real data. SPLSDA has two main advantages over SGPLS. First, it is computationally faster. The run time of SGPLS (which is not yet optimized for speed), including the tuning, is about 11 minutes for a sample size of $n = 100$ with $p = 1000$ predictors on a 64 bit machine with 3 GHz CPU. An SPLSDA run takes less than a minute for the same dataset. Second, since SPLSDA treats dimension reduction and classification in two separate steps, one has a wide choice of classifiers for its second stage. In the case of multicategory classification with unbalanced class sizes, SPLSDA might miss variables that discriminate classes with the smaller sample sizes from the rest of the classes. However, as the sample size increases, the variable selection performance of SPLSDA improves and it captures the true signals more accurately. As we have discussed in Section 4, current practice of PLS-based classification often involves variable filtering before PLSDA or GPLS. However, the choice of filtering methods and their tuning are still among open questions in high dimensional classification with PLS. The literature on variable filtering is rich with many pros and cons for both univariate and multivariate filtering approaches (Boulesteix et al. 2008). Boulesteix (2004) observed that PLSDA reached best classification accuracy with more than one PLS component and suggested that subsequent PLS components could be utilized for a better variable filtering. In a way, adapting SPLS for both PLSDA and GPLS is a principled move in this direction. We provide an implementation of SPLSDA and SGPLS as an R package at `http://cran.r-project.org/web/packages/spls`.

ACKNOWLEDGEMENTS

This research was supported by NSF grant DMS 0804597 and NIH grant HG03747 to SK.

APPENDIX: PROOFS

A.1. Proof of Theorem 1

Proof. After centering, the i-th row and $(g+1)$-th column of response matrix becomes

$$y^*_{i,(g+1)} = -\frac{n_g}{n} I(y_i \neq g) + \frac{n_{-g}}{n} I(y_i = g).$$

Then,

$$\begin{aligned} \boldsymbol{c^T X^T Y^*} &= \boldsymbol{c^T} \left[\tfrac{n_0 n_{-0}}{n} \begin{pmatrix} \hat{\mu}_{1,0} - \hat{\mu}_{1,-0} \\ \vdots \\ \hat{\mu}_{p,0} - \hat{\mu}_{p,-0} \end{pmatrix}, \;\cdots,\; \tfrac{n_G n_{-G}}{n} \begin{pmatrix} \hat{\mu}_{1,G} - \hat{\mu}_{1,-G} \\ \vdots \\ \hat{\mu}_{p,G} - \hat{\mu}_{p,-G} \end{pmatrix} \right] \\ &= \left[\frac{n_0 n_{-0}}{n} \sum_{j=1}^{p} c_j (\hat{\mu}_{j,0} - \hat{\mu}_{j,-0}), \cdots, \frac{n_G n_{-G}}{n} \sum_{j=1}^{p} c_j (\hat{\mu}_{j,G} - \hat{\mu}_{j,-G}) \right]. \end{aligned}$$

Hence,

$$\begin{aligned} \boldsymbol{\hat{c}} &= \text{argmax}_{\boldsymbol{c}} \boldsymbol{c^T X^T Y^* (Y^*)^T X c} - \lambda_1 \|\boldsymbol{c}\|_1 \\ &= \text{argmax}_{\boldsymbol{c}} \sum_{g=0}^{G} \left(\frac{n_g n_{-g}}{n} \right)^2 \left(\sum_{j=1}^{p} c_j (\hat{\mu}_{j,g} - \hat{\mu}_{j,-g}) \right)^2 - \lambda_1 \|\boldsymbol{c}\|_1 \end{aligned}$$

□

A.2. Proof of Theorem 2

Proof. After centering, the response vector becomes

$$y^*_i = -\frac{n_1}{n} I(y_i = 0) + \frac{n_0}{n} I(y_i = 1).$$

Consider the first direction vector of PLS without the sparsity penalty:

$$\boldsymbol{\hat{c}} = a_1 \boldsymbol{X^T Y}^*,$$

where a_1 is a positive constant. The j-th element of $\hat{\boldsymbol{c}}$ is then

$$\begin{aligned}
\hat{c}_j &= a_1 \sum_{i=1}^{n} x_{ij} y_i^* \\
&= a_1 \left(-\frac{n_1}{n} \sum_{i:y_i=0} x_{ij} + \frac{n_0}{n} \sum_{i:y_i=1} x_{ij} \right) \\
&= a_1 \left(-\frac{n_1}{n} n_0 \hat{\mu}_{j,0} + \frac{n_0}{n} n_1 \hat{\mu}_{j,1} \right) \\
&= a_1 \frac{n_0 n_1}{n} \left(\hat{\mu}_{j,1} - \hat{\mu}_{j,0} \right) \\
&= a_2 \left(\hat{\mu}_{j,1} - \hat{\mu}_{j,0} \right),
\end{aligned}$$

where $a_2 = a_1 n_0 n_1 / n$, which is a positive constant that does not depend on j, and x_{ij} is (i,j)-th element of $\boldsymbol{X}$. From Section 2, we have

$$\begin{aligned}
\hat{c}_j &= \left(|a_2(\hat{\mu}_{j,1} - \hat{\mu}_{j,0})| - \lambda_1/2 \right)_+ sign\left(a_2(\hat{\mu}_{j,1} - \hat{\mu}_{j,0}) \right) \\
&= a_2 \left(|\hat{\mu}_{j,1} - \hat{\mu}_{j,0}| - \lambda_1^* \right)_+ sign\left(\hat{\mu}_{j,1} - \hat{\mu}_{j,0} \right),
\end{aligned}$$

where $\lambda_1^* = \lambda_1 / 2a_2$.

For the second part of the theorem, note that

$$(n_0 + n_1 - 1) S_j^2 = (n_0 + n_1 - 2) S_{j,p}^2 + (\hat{\mu}_{j,1} - \hat{\mu}_{j,0})^2 n_0 n_1 / (n_0 + n_1),$$

where S_j^2 and $S_{j,p}^2$ are the overall and the pooled variances for j-th variable, respectively. If the columns of the predictor matrix are scaled to unit variance, then $S_j^2 = a_3$ and $S_{j,p}^2 = a_4 - a_5(\hat{\mu}_{j,1} - \hat{\mu}_{j,0})^2$ for positive constants a_3, a_4, and a_5 that do not depend on j. Then, the squared t-statistic for the j-th variable has the following form:

$$t_j^2 = \frac{(\hat{\mu}_{j,1} - \hat{\mu}_{j,0})^2}{S_{j,p}^2 (1/n_0 + 1/n_1)} = g((\hat{\mu}_{j,1} - \hat{\mu}_{j,0})^2),$$

for some strictly monotone function g. Hence,

$$\hat{c}_j = a \left(f(|t_j|) - \lambda_1^* \right)_+ sign\left(t_j \right), \quad j = 1, \cdots, p,$$

for some strictly monotone function f and a positive constant a. □

SUPPLEMENTAL MATERIALS

R package "spls": R package implementing the proposed methods. The package also contains the gene expression datasets used in the article. (GNU zipped tar file)

Additional datasets: Additional publicly available datasets that involve tumor classification with high dimensional gene expression data. (GNU zipped tar file)

Additional results: Application on the additional datasets. (PDF file)

References

Alizadeh, A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Powell, J. I., Yang, L., Marti, G. E., Moore, T., Jr., J. H., Lu, L., Lewis, D. B., Tibshirani, R., Sherlock, G., Chan, W. C., Greiner, T. C., Weisenburger, D. D., Armitage, J. O., Warnke, R., Levy, R., Wilson, W., Grever, M. R., Byrd, J. C., Botstein, D., Brown, P. O., , and Staudt, L. M. (2000), "Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling," *Nature*, 403, 503–511.

Barker, M. and Rayens, W. (2003), "Partial least squares for discrimination," *Journal of Chemometrics*, 17, 166–173.

Boulesteix, A.-L. (2004), "PLS dimension reduction for classification with microarray data," *Statistical Applications in Genetics and Molecular Biology*, 3, Article 33.

Boulesteix, A.-L., Strobl, C., T.Augustin, and Daumer, M. (2008), "Evaluating Microarray-based Classifiers: An Overview," *Cancer Informatics*, 6, 77–97.

Chun, H. and Keleş, S. (in press), "Sparse partial least squares for simultaneous dimension reduction and variable selection," *Journal of Royal Statistical Society, Series B*, (`http://www.stat.wisc.edu/~keles/Papers/SPLS_Nov07.pdf`).

Dettling, M. (2004), "BagBoosting for Tumor Classification with Gene Expression Data," *Bioinformatics*, 20, 3583–3593.

Dettling, M. and Bühlmann, P. (2002), "Supervised clustering of genes," *Genome Biology*, 3, research0069.1–0069.15.

Ding, B. and Gentleman, R. (2004), "Classification using generalized partial least squares," *Journal of Computational and Graphical Statistics*, 14, 280–298.

Firth, D. (1993), "Bias reduction of maximum likelihood estimates," *Biometrika*, 80, 27–38.

Fort, G. and Lambert-Lacroix, S. (2005), "Classification using partial least squares with penalized logistic regression," *Bioinformatics*, 21, 1104–1111.

Frank, I. E. and Friedman, J. H. (1993), "A statistical View of Some Chemometrics Regression Tools," *Technometrics*, 35, 109–135.

Friedman, J., Hastie, T., and Tibshirani, R. (2008), "Regularization paths for generalized linear models via coordinate descent," *Technical Report, Department of Statistics, Stanford University*, (`http://www-stat.stanford.edu/~hastie/Papers/glmnet.pdf`).

Marx, B. (1996), "Iteratively reweighted partial least squares estimation for generalized linear regression," *Technometrics*, 38, 374–381.

Nguyen, D. and Rocke, D. (2002a), "Muti-class cancer classification via partial least squares with gene expression profiles," *Bioinformatics*, 18, 1216–1226.

— (2002b), "Tumor classification by partial least squares using microarray gene expression data," *Bioinformatics*, 18, 39–50.

Singh, D., Febbo, P., Ross, K., Jackson, D., Manola, J., Ladd, C., Tamayo, P., Renshaw, A., D'Amico, A., Richie, J., Lander, E., Loda, M., Kantoff, P., Golub, T., and Sellers, W. (2002), "Gene expression correlates of clinical prostate cancer behavior," *Cancer Cell*, 1, 203–209.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002), "Diagnosis of multiple cancer types by shrunken centroids of gene expression," *Proceedings of the National Academy of Sciences*, 99, 6567–6572.

Wold, H. (1966), *Estimation of Principal Components and Related Models by Iterative Least Squares*, New York: Academic Press.