Boulesteix:

PLS dimension reduction for classification of microarray data

Sonderforschungsbereich 386, Paper 392 (2004)

Online unter: http://epub.ub.uni-muenchen.de/

PLS dimension reduction for classification of microarray data

Anne-Laure Boulesteix

July 5, 2004

Department of Statistics, University of Munich
Akademiestr.1, D-80799 Munich (Germany)
email: boulesteix@stat.uni-muenchen.de

Abstract

PLS dimension reduction is known to give good prediction accuracy in the context of classification with high-dimensional microarray data. In this paper, PLS is compared with some of the best state-of-the-art classification methods. In addition, a simple procedure to choose the number of components is suggested. The connection between PLS dimension reduction and gene selection is examined and a property of the first PLS component for binary classification is proven. PLS can also be used as a visualization tool for high-dimensional data in the classification framework. The whole study is based on 9 real microarray cancer data sets.

1 Introduction

The output of n microarray experiments can be summarized as a $n \times p$ data matrix, where p is the number of analyzed genes. p is always much larger than the number of experiments n. An important application of microarray technology is tumor diagnosis, i.e. class prediction. High-dimensionality makes the application of most classifi cation methods diffi cult, if not impossible.

To overcome this problem, one can either extract a small subset of interesting variables (gene selection) or construct m new components which summarize the original data as well as possible, with $m < p$ (dimension reduction).

Gene selection has been studied extensively in the last few years. The most commonly used gene selection procedures are based on a score which is calculated for all genes individually. Then the genes with the best scores are selected. These methods are often denoted as univariate gene selection. Several selection criteria have been used in the literature, e.g. the t statistic (Hedenfalk et al., 2001), the Wilcoxon's rank sum statistic (Dettling and Bühlmann, 2003) or Ben Dor's combinatoric 'TNoM' score (Ben-Dor et al., 2000). When using a test statistic as criterion, it is useful to adjust the p-values with a multiple testing procedure (Dudoit et al., 2003). The main advantages of gene selection are its simplicity and its interpretability. Gene selection procedures output a list of relevant genes which can be experimentally analyzed by biologists afterwards. Moreover, univariate gene selection is generally very fast.

However, a large part of the information contained in the data gets lost when genes are selected solely according to their individual capacity to separate the classes. Interactions and correlations between genes are omitted, although they are of great interest in system biology. A few sophisticated procedures intend to overcome this problem by selecting optimal subsets with respect to a given criterion instead of ranking the genes. Bo and Jonassen (2002) look for relevant pairs of genes, whereas Li et al. (2001) want to find optimal gene subsets via genetic algorithms. However, these methods generally suffer from overfitting: the obtained gene subsets might be optimal for the training data, but they do not perform as well on independent test data. Moreover, they are based on computationally intensive iterative algorithms and thus very difficult to interpret and implement.

Dimension reduction is a wise alternative to variable selection in order to overcome this dimensionality problem. It is also denoted as feature extraction. Unlike gene selection, such methods use all the genes included in the data set. The whole data are projected onto a low-dimensional space, thus allowing a graphical representation. The new components often give information or hints about the data's intrinsic structure, although there is no standard concept and procedure to do this. Dimension reduction is sometimes criticized for its lack of interpretability, especially for applied scientists who often need more concrete answers about indi-

vidual genes. In this paper, we show that PLS dimension reduction is tightly connected to gene selection.

Dimension reduction methods for classification can be categorized into linear and nonlinear, supervised and unsupervised methods. Intuitively, supervised methods, i.e. methods which use the class information of the observations to construct new components, should be preferred to unsupervised methods, which work only 'by chance' in 'good' data sets (Nguyen and Rocke, 2002). Since nonlinear methods are generally computationally intensive and lack of robustness, they are not recommended for microarray data analysis. To our knowledge, the only well-established supervised linear dimension reduction method working even if $n < p$ is the Partial Least Squares method (PLS). PLS is a linear method in the sense that the new components are linear combinations of the original variables. However, the coefficients defining the new components are not linear. Another approach denoted as between-group analysis has been proposed by Culhane et al. (2002), but it turns out that it is strongly related to PLS. Principal component analysis (Ghosh, 2002; Kahn et al., 2001) is an unsupervised method. As such, it is inappropriate for classification. Other methods, such as sufficient dimension reduction (Chiaromonte and Martinelli, 2001) generally require preliminary feature selection and miss potentially interesting information.

It is well-known that PLS dimension reduction is much faster than gene selection and leads to very accurate classification (Nguyen and Rocke, 2002; Huang and Pan, 2003). However, these papers do not include any extensive comparative study of classifiction methods. Moreover, they treat the PLS technique as a 'black box' which is only meant to improve classification accuracy, without concern for the components themselves. In this paper, three aspects of PLS dimension reduction are examined. First, how does it perform in comparison with the top-ranking classification methods which have already been studied in the literature ? Second, can PLS dimension reduction be used for gene selection ? Third, is PLS useful for visualization and interpretation of the data's structure ?

In recent years, aggregation methods such as bagging (Breiman, 1996) and boosting (Freund, 1995) have been extensively analyzed. They lead to spectacular improvements of prediction accuracy when they are applied to classification problems. In microarray data analysis, accuracy improvement is also observed (Dettling and Bühlmann, 2003; Dudoit et al., 2002), although not

as spectacular. So far, aggregating methods have been applied in association with weak and unstable classifiers like stumps or classification trees. To our knowledge, boosting has never been used with dimension reduction techniques. In this paper, we perform classification using PLS dimension reduction and apply a boosting algorithm to this method.

The paper is organized as follows. PLS dimension reduction and boosting are introduced in section 2. Classification results using PLS and PLS with boosting are presented in section 3. In section 4, the connection between PLS and gene selection is studied and an interesting property of the first PLS component is proved in the case of binary responses. Section 5 shows how PLS dimension reduction can be used for visualization of subclasses.

In the following, $X_1, \ldots, X_p$ denote the continuous predictors (genes) and $\mathbf{x} = (X_1, \ldots, X_p)^T$ the corresponding random vector. $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})^T$ for $i = 1, \ldots, n$ denote independent identically distributed realizations of the random vector $\mathbf{x}$. Each row of the $n \times p$ data matrix $\mathbf{X} \in \mathbb{R}^{n \times p}$ contains a realization of $\mathbf{x}$.

2 Dimension reduction and classification with PLS

2.1 Introduction to PLS regression

The method denoted as Partial Least Squares (PLS) was originally developed as a mmultivariate regression tool in the context of chemometrics. An overview of the history of PLS regression is given in (Martens, 2001). PLS regression is especially appropriated to predict a univariate or multivariate continuous response using a large number of continuous predictors.

Suppose we have a $n \times p$ data matrix $\mathbf{X}$. The centered data matrix $\mathbf{X}_C$ is obtained by centering each column to zero mean. In section 2.1, Y denotes a univariate continuous reponse variable and $\mathbf{Y}$ the $n \times 1$ vector containing the realizations of Y for the n observations. The centered vector $\mathbf{Y}_C$ is obtained by substracting the empirical mean of Y from $\mathbf{Y}$.

PLS was first developed as an algorithm performing matrix decompositions. In (Helland, 1988), PLS regression is formulated as follows. $\mathbf{X}_C$ and $\mathbf{Y}_C$ are simultaneously decomposed

into underlying components $\mathbf{t}_1, \ldots, t_m$:

$$\mathbf{X}_C = \mathbf{t}_1\mathbf{p}_1^T + \cdots + \mathbf{t}_m\mathbf{p}_m^T + \mathbf{E}_m \tag{1}$$

$$\mathbf{Y}_C = \mathbf{t}_1 q_1 + \cdots + \mathbf{t}_m q_m + \mathbf{f}_m, \tag{2}$$

where m is the chosen number of components. $\mathbf{t}_1, \ldots, \mathbf{t}_m \in \mathbb{R}^n$ represent the n observations of the m components. They are usually denoted as scores. $\mathbf{p}_1, \ldots, \mathbf{p}_m \in \mathbf{R}^p$ and $q_1, \ldots, q_m \in \mathbb{R}$ are usually denoted as loadings. $\mathbf{E}_m \in \mathbb{R}^{n \times p}$ and $\mathbf{f}_m \in \mathbb{R}^n$ are residuals. The underlying components $\mathbf{t}_1, \ldots, \mathbf{t}_m \in \mathbb{R}^n$ are linear combinations of the original variables $X_1, \ldots, X_p$, i.e.

$$\mathbf{t}_1 = \mathbf{X}_C\mathbf{a}_1$$

$$\ldots$$

$$\mathbf{t}_m = \mathbf{X}_C\mathbf{a}_m,$$

where $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^p$ have to be computed by an algorithm. It is easy to see that the $\mathbf{t}_1, \ldots, \mathbf{t}_m$, $\mathbf{p}_1, \ldots, \mathbf{p}_m$ and $q_1, \ldots, q_m$ are not unique. Thus, one has to adopt restrictions, for instance an orthogonality constraint. The most commonly used algorithm for univariate PLS regression outputs orthogonal components, i.e. $\mathbf{t}_i^T\mathbf{t}_j \forall i \neq j$, see e.g. (Martens and Naes, 1989). An alternative algorithm which outputs orthogonal loadings can be found in (Naes et al., 1985). It can be shown that both algorithms yield the same prediction if a linear model is built using the latent variables as predictors (Helland, 1988).

Later on, PLS regression was studied by statisticians (Stone and Brooks, 1990; Garthwaite, 1994; Frank and Friedman, 1993). It turns out that the algorithm with orthogonal components can be interpreted in terms of an optimality criterion based on the empirical covariance of $\mathbf{x}$ and Y. In (Stone and Brooks, 1990), $\mathbf{a}_1, \ldots, \mathbf{a}_m$ are defined as follows.

Definition 1 *Let $\hat{COV}$ denote the empirical covariance and $\hat{\mathbf{\Sigma}}$ the empirical covariance matrix of $\mathbf{x}$. $\mathbf{a}_1$ is the unit vector (i.e. $\mathbf{a}_1^T\mathbf{a}_1 = 1$) maximizing $\hat{COV}(\mathbf{a}_1^T\mathbf{x}, y) = (\mathbf{X}_C\mathbf{a}_1)^T\mathbf{Y}_C$. $\mathbf{a}_2$ is the unit vector maximizing $\hat{COV}(\mathbf{a}_2^T\mathbf{x}, y)$ subject to the constraint $\mathbf{a}_2^T\hat{\mathbf{\Sigma}}\mathbf{a}_1 = 0$, and so on,*

Stone and Brooks (1990) show that the algorithm with orthogonal factors computes the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m$ as defined in Definition 1.

For a multivariate response $\mathbf{y} \in \mathbb{R}^q$, $\mathbf{Y}$ has the form of a $(n \times q)$ data matrix, where q is the number of responses. $\mathbf{Y}_C$ denotes the matrix obtained from $\mathbf{Y}$ by centering the columns to zero

mean. The so-called SIMPLS algorithm proposed by de Jong (1993) was developed to satisfy an optimality criterion. It computes the vectors $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^p$ and $\mathbf{b}_1, \ldots, \mathbf{b}_m \in \mathbb{R}^q$ defined as follows.

Definition 2 *$\mathbf{a}_1$ and $\mathbf{b}_1$ are the unit vectors maximizing $C\hat{O}V(\mathbf{a}_1^T\mathbf{x}, \mathbf{b}_1^T\mathbf{y})$. $\mathbf{a}_2$ and $\mathbf{b}_2$ are the unit vectors maximizing $C\hat{O}V(\mathbf{a}_2^T\mathbf{x}, \mathbf{b}_2^T\mathbf{y})$ subject to the constraint $\mathbf{a}_2^T\hat{\mathbf{\Sigma}}\mathbf{a}_1 = 0$, and so on.*

The SIMPLS algorithm is based on the singular value decomposition of a $p \times q$ matrix which is set to $\mathbf{S} = \mathbf{X}^T\mathbf{Y}$ in the first iteration. An implementation of the SIMPLS algorithm is included in the `R` library `pls.pcr`. Since the SIMPLS algorithm and the algorithm with orthogonal components are equivalent for univariate responses, only the SIMPLS algorithm is used in this paper.

2.2 PLS and dimension reduction in the classification framework

From now on, Y denotes a categorical variable taking values 1 to K, with $K \geq 2$. $Y_1, \ldots, Y_n$ denote the n realizations of Y. In this framework, PLS can be seen as a dimension reduction method: $\mathbf{t}_1, \ldots, \mathbf{t}_m \in \mathbb{R}^n$ represent the observed m new components. Although the algorithm with orthogonal components has been designed for continuous responses, it is known to lead to good classification accuracy when it is applied to a binary response ($K = 2$), especially for high-dimensional data as microarray data (Nguyen and Rocke, 2002; Huang and Pan, 2003). The same can be said for the SIMPLS algorithm: a binary response can be treated as a continuous response, since no distributional assumption is necessary to use the SIMPLS algorithm.

If the response is multicategorical ($K > 2$), it can not be treated as a continuous variable. The problem can be circumvented by dummy coding. The multicategorical random variable Y is transformed into a K-dimensional random vector $\mathbf{y} \in \{0, 1\}^K$ as follows.

$$y_{i1} = 1 \quad \text{if } Y_i = k,$$
$$y_{ik} = 0 \quad \text{else},$$

where $\mathbf{y}_i = (y_{i1}, \ldots, y_{iK})^T$ denotes the ith realization of $\mathbf{y}$. $\mathbf{Y}$ denotes the $n \times K$ matrix containing $\mathbf{y}_i$ in its ith row, for $i = 1, \ldots, n$.

In the following, $\mathbf{Y}$ denotes the $n \times 1$ vector $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ if Y is binary ($K = 2$) or the $n \times K$ matrix as defined above if Y is multicategorical ($K > 2$). In both cases, the

SIMPLS algorithm outputs a $p \times m$ transformation matrix $\mathbf{A}$ containing the $\mathbf{a}_1, \ldots, \mathbf{a}_m \in \mathbb{R}^p$ in its columns. The $n \times m$ matrix $\mathbf{T}$ containing the values of the new components for the n observations is computed as

$$\mathbf{T} = \mathbf{X}_C \mathbf{A}.$$

These new components can be used as predictors for classification. WhereasHuang and Pan (2003) build a classical linear model to predict the class y, Nguyen and Rocke (2002) use logistic regression and linear discriminant analysis. See (Hastie et al., 2001) for an overview of classical classification methods. In this paper, we use linear discriminant analysis, because logistic regression performs sometimes poorly on 'good' data sets due to convergence problems.

The classification method described above can be formalized as follows. A denotes the function of $\mathbf{X}$ and $\mathbf{Y}$ which outputs the PLS transformation matrix:

$$\begin{array}{rccl} A: & \mathbb{R}^n \times \mathbb{R}^p \times \{1, \ldots, K\}^n & \rightarrow & \mathbb{R}^p \times \mathbb{R}^m \\ & (\mathbf{X}, \mathbf{Y}) & \rightarrow & \mathbf{A}. \end{array}$$

$\delta_{LDA}(., \mathbf{X}, \mathbf{Y})$ denotes the linear discriminant function which predicts the class of observation $\mathbf{x}_{new}$ based on the matrix of predictors $\mathbf{X}$ and the response vector or matrix $\mathbf{Y}$. The classification method consisting of dimension reduction using PLS and linear discriminant analysis using the obtained components can be summarized using the classical representation of a discriminant function:

$$\begin{array}{rccl} \delta_{PLS}(., \mathbf{X}, \mathbf{Y}): & \mathbb{R}^p & \rightarrow & \{1, \ldots, K\} \\ & \mathbf{x}_{new} & \rightarrow & \delta_{LDA}(A(\mathbf{X}, \mathbf{Y})^T \mathbf{x}_{new}, \mathbf{X}A(\mathbf{X}, \mathbf{Y}), \mathbf{Y}), \end{array}$$

where the vector $\mathbf{x}_{new}$ has already been centered by substracting the empirical mean vector of $\mathbf{x}$ calculated from $\mathbf{X}$.

2.3 Choosing the number of components

There is no widely accepted procedure to determine the right number of PLS components. Here, we propose a simple method based on cross-validation. Only the learning set $\mathcal{L}$ is used to choose m. The following procedure is repeated N_{run} times: the classifier δ_{PLS} is built using only $\alpha\%$ of the observations from $\mathcal{L}$ and applied to the remaining observations, with m taking successively different values. After N_{run} runs, the mean error rate is computed for each value

of m. The value of m minimizing the error rate is chosen. In our analysis, we set α to 0.7 and N_{run} to 50.

2.4 Boosting

Bagging and boosting consist of building a simple classifier using successively different bootstrap samples. In bagging, the bootstrap samples are based on the unweighted bootstrap and the predictions are made by majority voting. In boosting, the bootstrap samples are built iteratively using weights that depend on the predictions made in the last iteration. An early study focusing on statistical aspects of boosting is (Schapire et al., 1998). A classifier based on a learning set $\mathcal{L}$ containing n_L observations is represented as in the previous section as a function of the p-dimensional vector of predictors $\mathbf{x}_{new}$:

$$\begin{aligned} C(., \mathbf{X}_L, \mathbf{Y}_L): \quad \mathbb{R}^p \quad &\rightarrow \quad \{1, \ldots, K\} \\ \mathbf{x}_{new} \quad &\rightarrow \quad C(\mathbf{x}_{new}, \mathbf{X}_L, \mathbf{Y}_L), \end{aligned}$$

where the index L means that only observations from the learning set $\mathcal{L}$ are included in the matrices $\mathbf{X}_L$ and Y_L. In boosting, perturbed learning sets $\mathcal{L}_1, \ldots, \mathcal{L}_B$ are formed adaptively by drawing from the learning set $\mathcal{L}$ at random, where the probability of an observation to be selected in $\mathcal{L}_k$ depends on the prediction made by $C(., \mathbf{X}_{L_{k-1}}, \mathbf{Y}_{L_{k-1}})$. Observations which are wrongly classified by $C(., \mathbf{X}_{L_{k-1}}, \mathbf{Y}_{L_{k-1}})$ have greater probability to be selected in $\mathcal{L}_k$.

The discrete AdaBoost procedure was proposed by Freund (1995). In the first iteration, the weights are initialized to $w_1 = \cdots = w_{n_L} = 1/n_L$. In the following we show the k-th step of the algorithm as described by Tutz and Hechenbichler (2004).

Discrete AdaBoost algorithm

1. - Based on the resampling probabilities $w_1, \ldots, w_{n_L}$, the learning set $\mathcal{L}_k$ is sampled from $\mathcal{L}$ with replacement.
   - The classifier $C(., \mathbf{X}_{L_k}, \mathbf{Y}_{L_k})$ is built.

2. The learning set $\mathcal{L}$ is run through the classifier $C(., \mathbf{X}_{L_k}, \mathbf{Y}_{L_k})$ yielding an error indicator $\epsilon_i = 1$ if the i-th observation is classified incorrectly and $\epsilon_i = 0$ otherwise.

3. With $e_k = \sum_{i=1}^{n_L} w_i \epsilon_i$, $b_k = (1 - e_k)/e_k$ and $c_k = \log(b_k)$ the resampling probabilities are updated for the next step by

$$w_{i,new} = \frac{w_i b_k^{\epsilon_i}}{\sum_{j=1}^{n_L} w_j b_k^{\epsilon_j}} = \frac{w_i \exp(c_k \epsilon_i)}{\sum_{j=1}^{n_L} w_j \exp(c_k \epsilon_j)}$$

After B iterations the aggregated voting for observation $\mathbf{x}_{new}$ is obtained by

$$\arg\max_j (\sum_{k=1}^{B} c_k I(C(x, \mathbf{X}_{L_k}, \mathbf{Y}_{L_k}) = j))$$

In this paper, we propose to apply the AdaBoost algorithm with $C = \delta_{PLS}$.

3 Classification results on real microarray data

3.1 Data sets

Colon: The colon data set is a publicly available 'benchmark' gene expression data set which is extensively described in (Alon et al., 1999). The data set contains the expression levels of 2000 genes for 62 patients from two classes. 22 patients are healthy patients and 40 patients have colon cancer.

Leukemia Data: This data set was introduced in (Golub et al., 1999) and contains the expression levels of 7129 genes for 47 ALL-leukemia patients and 25 AML-leukemia patients. It is included in the `R` library `golubEsets`. After data preprocessing following the procedure described in (Dudoit et al., 2002), only 3571 variables remain. It is easy to achieve excellent classification accuracy on this data set, even with quite trivial methods as described in the original paper (Golub et al., 1999).

Prostate: This data set gives the expression levels of 12600 genes for 50 normal tissues and 52 prostate cancer tissues. We threshold the data and filter genes as described in (Singh et al., 2002). The filtering step leaves us with 5908 genes.

Breast cancer (ER+/ER-): This data set gives the expression levels of 7129 genes for 46 breast cancer patients from which 23 have status ER+ and 23 have status ER-. It is presented in (West et al., 2001).

Carcinoma: This dataset comprises the expression levels of 7463 genes for 18 normal tissues and 18 carcinomas. We standardize each array to zero mean and unit variance. For an extensive description of the data set, see (Notterman et al., 2001).

Lymphoma: The dataset presented by Alizadeh et al. (2000) comprises the expression levels of 4026 genes for 62 patients from 3 different classes (B-CLL, FL and DLBCL). We inputed the missing values as described in (Dudoit et al., 2002) using the function `pamr.inpute` from the `R` library `pamr`.

SRBCT microarray data: This gene expression data set is presented in (Kahn et al., 2001). It contains the expression levels of 2308 genes for 83 Small Round Blue Cells Tumor (SRBCT) patients belonging to one of the 4 tumor classes: Ewing family of tumors (EWS), non-Hodgkin lymphoma (BL), neuroblastoma (NB) and rhabdomyosarcoma (RMS).

Breast cancer (BRCA): This breast cancer data set contains the expression levels of 3227 genes for breast cancer patients with one of the three tumor types: sporadic, BRCA1 and BRCA2. It is described in (Hedenfalk et al., 2001). The data are preprocessed as described in (Simon et al., 2004).

NCI: This dataset comprises the expression levels of 5244 genes for 61 patients with 8 different tumor types: 7 breast, 5 central nervous system, 6 ovarian, 7 colon, 6 leukemia, 8 melanoma, 9 non-small-cell-lung-carcinoma, 6 ovarian, 9 renal (Ross et al., 2000). The data are preprocessed as described in (Dudoit et al., 2002).

3.2 Study design

For each data set, 200 random partitions into a learning data set $\mathcal{L}$ containing n_L observations and a test data set $\mathcal{T}$ containing the $n-n_L$ remaining observations are generated. This approach for evaluating classifi cation methods was used in one of the most extensive comparative studies of classifi cation methods for microarray data (Dudoit et al., 2002). It is believed to be more reliable than cross-validation (Braga-Neto et al., 2004). We fi x the ratio n_L/n at 0.7, which is a usual choice. For each partition $\{\mathcal{L}, \mathcal{T}\}$, we predict the class of the observations from $\mathcal{T}$ using δ_{PLS} with successively 1,2,3,4,5 PLS components for the data sets with binary responses. We also use the discrete AdaBoost boosting algorithm based on the classifi er $C = \delta_{PLS}$ with 1,2,3,4 PLS components. For data sets with multicategorical responses, we use 1,2,3,4,5,6 PLS compo-

nents for the lymphoma and BRCA data, 1,2,3,4,5,6,8,10 for the SRBCT data and 1,5,10,15,20 components for the NCI data. For each approach and for each number of components, the mean error rate over the 200 partitions is computed. The results are summarized in tables.

For each partition $\{\mathcal{L}, \mathcal{T}\}$, the optimal number of PLS components m_{opt} is estimated following the procedure described in section 2.3 and the error rate of δ_{PLS} with m_{opt} PLS components is computed. The corresponding mean error rate over the 200 random partitions is given in the table of results.

For comparison, the mean error rate obtained with some of the best classification methods for microarray data is also computed. The first one is nearest-neighbor classification based on 5 neighbors (5NN). The second one is linear discriminant analysis (LDA), as described in (Dudoit et al., 2002). These two methods are known to achieve excellent classification accuracy (Dudoit et al., 2002). The third one is Support Vector Machines (SVM). This method is used e.g. by Furey et al. (2000) and seems to perform well on microarray data. For an overview of classical classification methods, see (Hastie et al., 2001). 5NN, LDA and SVM require preliminary gene selection. The gene selection is performed by ranking genes according to the BSS/WSS-statistic, where BSS denotes the between-group sum of squares and WSS the within-group sum of squares. For gene j the BSS/WSS-statistic is calculated as

$$BSS_j/WSS_j \;=\; \frac{\sum_{k=1}^{K}\sum_{i:Y_i=k}(\hat{\mu}_{jk}-\hat{\mu}_j)^2}{\sum_{k=1}^{K}\sum_{i:Y_i=k}(x_{ij}-\hat{\mu}_{jk})^2},$$

where $\hat{\mu}_j$ is the sample mean of X_j and $\hat{\mu}_{jk}$ is the sample mean of X_j within class k, for $k = 1, \ldots, K$. The genes with the highest BSS/WSS-statistic are selected. There is no golden rule to choose the number of genes to select. In this study, we decide to use 20 or 50 genes for data sets with binary responses and 100 and 200 genes for data sets with multicategorical responses. These numbers are in agreement with similar studies found in the literature, e.g. (Dudoit et al., 2002). At last, we apply a recent method called 'prediction analysis of microarray'(PAM) which was especially designed for high-dimensional microarray data (Tibshirani et al., 2002). To our knowledge, it is the only fast classification method beside PLS which can be applied to high-dimensional data without gene selection. PAM is based on shrunken centroids and necessitates the choice of the shrinkage parameter δ. The number of genes used to compute the shrunken centroids depends on δ. A possible choice is $\delta = 0$: all genes are used to compute the centroids.

Tibshirani et al. (2002) propose to select the best value of δ by cross-validation. In our study, we try successively both approaches: $\delta = 0$ (denoted as PAM) and $\delta = \delta_{opt}$ (denoted as PAM-opt), where δ_{opt} is determined by cross-validation. The PAM method is implemented in the R library `pamr`.

The table of results contains only the error rates obtained with 5NN, SVM, PAM and PAM-opt, because the classification accuracy with LDA was found to be comparatively bad for all data sets. The number of selected genes is specified for each method: for example, 'SVM-20' means Support Vector Machines with 20 selected genes.

3.3 Classification accuracy of δ_{PLS}

The results are summarized in Table 1. The data sets with binary responses can be divided in two groups. For the leukemia and carcinoma data, the classification accuracy does not depend much on the number of PLS components. It seems that subsequent components are only noise. On the contrary, the error rate is considerably reduced by using more than one component for the colon, prostate and breast cancer data. The improvement is rather dramatic for the prostate data. Thus, it seems that for data sets with low error rates (leukemia, carcinoma), the classes are optimally separated by one component, whereas subsequent components are useful for data sets with high error rates (prostate, colon, breast cancer). For all 5 data sets, the classification accuracy is excellent compared to the other methods.

Moreover, PLS dimension reduction is very fast because it is based on linear operations with small matrices. The proposed procedure is much faster than the standard approach consisting of selecting a gene subset and building a classifier on this subset. For the lymphoma data and the SRBCT data, $K - 1$ seems to be the minimum number of PLS components required to obtain a good classification accuracy. It is noticeable that δ_{PLS} can also perform very well on data sets with many classes ($K = 8$ for the NCI data).

As can be seen from Table 1, the number of components giving the best classification accuracy is not the same for all data sets. When our procedure to determine the number of useful PLS components is used for each partition $(\mathcal{L}, \mathcal{T})$, the classification accuracy turns out to be quite good, although not as good as the accuracy obtained with the best number of components identified a posteriori from the table of results. In our study including 200 random partitions,

the number of runs in the estimation procedure was set at $N_{run} = 50$ for computational reasons, but a biologist working with one learning set and one test set could perform more runs, which would make the procedure more reliable. In Figure 1, histograms of m_{opt} over the 200 random partitions are represented for each data set. These histograms agree with Table 1. For instance, the most frequent value of m_{opt} for the colon data is data is 2. It can be seen in Table 1 that the best classification accuracy is obtained with 2 PLS components for the colon data.

Some of the classical methods tested in this paper also perform well, especially SVM and PAM. The performance of SVM is slightly better. However, a pitfall of SVM is that it necessitates gene selection in practice, although not in theory. On the whole, the PLS-based method presented in this paper performs better than all the other methods for most data sets, as can be seen from Figure 2. This accuracy is not reached at the expense of computational time. PLS is a fast efficient which did never fail to give a good to excellent classification accuracy for all the studied data sets. Since the best number of components can be estimated by cross-validation, the method does not involve any 'free' parameter like the number of selected genes for SVM.

3.4 Classification accuracy of discrete AdaBoost with $C = \delta_{PLS}$

In this section, we compute the mean classification error rate over the 200 random partitions using the AdaBoost algorithm with $C = \delta_{PLS}$ and $B = 30$. $B = 30$ turns out to be a sensible choice, because the classification accuracy remains constant after approximately 20 iterations. The results are represented in Figure **??** for the prostate data. Boosting can reduce the error rate when one or two PLS components are used. It can be seen from Table 1 that the best classification accuracy for δ_{PLS} is reached with three PLS components: the fourth and fifth PLS components do not improve the classification accuracy. It suggests a similarity between boosting and PLS.

This fact can be intuitively explained as follows. At iteration k in boosting, an observation is either in or out of the learning set, and the probability depends on how the observation was classified at iteration $k-1$. In PLS, each observation plays a part in the construction of the kth PLS component but only the residuals of the model at iteration $k-1$ are used. Consequently, observations which would be wrongly classified with $k-1$ PLS components play a more important part in the construction of the k-th component, like in boosting.

For the colon, leukemia and carcinoma data, boosting does not improve the classification accuracy of δ_{PLS}. Thus, we focus on the prostate data in the following. In order to examine the connection between boosting and PLS, we perform dimension reduction by PLS on the whole prostate data set. We also run the AdaBoost algorithm with $C = \delta_{PLS}$ (1 component) and compute the empirical correlations between the 4 first PLS components and the first component obtained at each boosting iteration. The results are shown for 5 boosting iterations in Table 3. The first component at each boosting iteration is strongly correlated with the first and the second PLS component, but not with the subsequent components. This statement agrees with the classification accuracy results: it can be seen from Figure **??** that the classification accuracy obtained by boosting with one component equals approximately the classification accuracy of δ_{PLS} with two components. The study of the theoretical relationship between PLS and boosting could be examined in further work.

In the following section, we show a connection between the first PLS component and gene selection: the squared coefficient in the first PLS component can be seen as a score of relevance for single genes (see section 4 for more details). Boosting with the classifier δ_{PLS} can thus be seen as a kind of 'boosted gene selection'. We suggest that selecting the top-ranking genes at each boosting iteration might improve the classification accuaracy of classifiers based on small gene subsets, although the study of this topic would be beyond the scope of this paper.

4 PLS and gene selection

Biologists often want statisticians to answer questions like 'which genes can be used for tumor diagnosis ?'. Thus, gene selection remains an important issue and should not be neglected. Dimension reduction is sometimes wrongly described as a black box which looses the information about single genes. In the following, we will see that PLS performs gene selection intrinsically.

In this section, only binary responses are considered: Y can take values 1 and 2. We denote as $\mathbf{Y}_C = (Y_{C1}, \ldots, Y_{Cn})^T$ the vector obtained by centering $\mathbf{Y} = (Y_1, \ldots, Y_n)^T$ to zero mean:

$$\begin{aligned} Y_{Ci} &= -n_2/n \quad \text{if } Y_i = 1, \\ &= \quad n_1/n \quad \text{if } Y_i = 2, \end{aligned}$$

where n_1 resp. n_2 are the numbers of observations in class 1 resp. 2.

To perform PLS dimension reduction, it is not necessary to scale each column of the data matrix $\mathbf{X}$ to unit variance. However, the first PLS component satisfies an interesting property with respect to gene selection if $\mathbf{X}$ is scaled. In this section, the columns of the data matrix $\mathbf{X}$ are supposed to be have been scaled to unit variance and, as usual in the PLS framework, centered to zero mean. $\mathbf{a} = (a_1, \ldots, a_p)^T$ denotes the $p \times 1$ vector defining the first PLS component as calculated by the SIMPLS algorithm.

A classical gene selection scheme consists of ordering the p genes according to BSS_j/WSS_j and selecting the top-ranking genes. For data sets with binary responses, we argue that a_j^2 can also be seen as a scoring criterion for gene j and we proove that the ordering of the genes obtained using BSS_j/WSS_j is the same as the ordering obtained using a_j^2.

Theorem 1 *If $K = 2$, there exists a strictly monotonic function f such that*

$$BSS_j/WSS_j = f(a_j^2),$$

for $j = 1, \ldots, p$.

Proof. From the SIMPLS algorithm, we get

$$\mathbf{a} = c_1 \cdot \mathbf{X}^T \mathbf{Y}_C,$$

where c_1 is a scalar. For $j = 1, \ldots, p$,

$$a_j = c_1 \cdot \sum_{i=1}^{n} x_{ij} Y_{Ci}.$$

It leads to

$$\begin{aligned} a_j &= c_1 \cdot (-(n_2/n) \textstyle\sum_{i:Y_i=1} x_{ij} + (n_1/n) \sum_{i:Y_i=2} x_{ij}) \\ a_j^2 &= c_1^2 \cdot (n_1 n_2/n)^2 (\hat{\mu}_{j2} - \hat{\mu}_{j1})^2 \end{aligned}$$

For $K = 2$,

$$\begin{aligned} BSS_j &= n_1(\hat{\mu}_{j1} - \hat{\mu}_j)^2 + n_2(\hat{\mu}_{j2} - \hat{\mu}_j)^2 \\ &= n_1((n\hat{\mu}_{j1} - n_1\hat{\mu}_{j1} - n_2\hat{\mu}_{j2})/n)^2 + n_2((n\hat{\mu}_{j2} - n_2\hat{\mu}_{j2} - n_1\hat{\mu}_{j1})/n)^2 \\ &= (n_1 n_2^2/n^2 + n_2 n_1^2/n^2)(\hat{\mu}_{j2} - \hat{\mu_{j1}})^2 \\ &= c_2 a_j^2, \end{aligned}$$

where c_2 is a positive constant which does not depend on j. $BSS_j + WSS_j$ is proportional to the sample variance of X_j. Since the variables $X_1, \ldots, X_p$ all have equal sample variance, there exists a constant c_3 which is independent of j such that

$$\begin{aligned} BSS_j/WSS_j &= \frac{BSS_j}{c_3 - BSS_j} \\ &= \frac{c_2 a_j^2}{c_3 - c_2 a_j^2}. \end{aligned}$$

□

As a consequence, the first PLS component calculated by the SIMPLS algorithm can be used to order and select genes and the ordering is the same as the ordering produced by one of the most widely accepted selection criteria. Up to a constant, the BSS/WSS-statistic equals the F-statistic which is used to test the equality of the means within different groups. Since BSS/WSS is obtained by a strictly monotonic transformation of a_j^2, a_j^2 can be seen as a test statistic itself. This PLS-based procedure for gene selection is much faster than the computation of BSS/WSS for each gene.

5 Visualization and subclasses

An other advantage of PLS dimension reduction is the possibility to visualize the data by graphical representation. For instance, one can plot the second PLS component against the first PLS component using different colors for each class. As a visualization method, PLS might be useful for applied researchers who need simple graphical tools. However, a question remains open: are the PLS components only by-products of the classification method or can they be interpreted, for instance in terms of clusters ? In the following, we address this question.

Suppose we have to analyse a data set with binary response. One of the classes, e.g. class 2, consists of 2 subclasses: 2a and 2b. In the following, we try to interpret the PLS components in terms of clusters. For example, the first PLS component may discriminate between class 1 and class 2a and the second PLS component between class 1 and class 2b. In order to illustrate this point, we perform PLS dimension reduction on the whole prostate data set. We also cluster the observations from class 2 into two subclasses 2a and 2b using the k-means algorithm on the original variables $X_1, \ldots, X_p$. As can be seen from Figure 3, the first PLS component separates almost perfectly class 1 and class 2b, whereas the second PLS component separates almost

perfectly class 1 and class 2a. Thus, the two PLS components can be interpreted in terms of clusters.

A similar result can be obtained with the breast cancer data. We perform PLS dimension reduction on the whole breast cancer data set and cluster the observations from class 2 into 2a and 2b using the k-means algorithm on $X_1, \ldots, X_p$. The first and the second PLS components are reprensented as a scatterplot in Figure 4. We observe that the first PLS component can separate class 1 from class 2 perfectly. The second PLS component separates only 1 and 2a from 2b. Similar results are observed for the carcinoma and the leukemia data. Thus, for 4 of 5 data sets with binary class, the PLS components can be easily interpreted in terms of clusters.

However, in our examples, we do not know whether the subclasses 2a and 2b are biologically interpretable: they are only the output of the k-means clustering algorithm. Thus, we also perform the same analysis on the lymphoma data set, for which we know three biologically interpretable classes. Patients with tumor type DLBCL are assigned to class 1, B-CLL to class 2a and FL to class 2b. We perform PLS dimension reduction as if the class were binary. As can be seen from Figure 5, the first PLS discriminates between class 1 and class 2, whereas the second PLS discriminates between class 2a and classes 1 and 2b.

As a conclusion, we recommend the PLS technique as a visualization tool, because it can outline relevant cluster structures.

6 Discussion

In this paper, several aspects of PLS dimension reduction for classification are examined. First, PLS is compared to several other classification methods which are known to give excellent classification accuracy. To our knowledge, this work is the first extensive comparison study including PLS. The classifier δ_{PLS} turns out to be the best one in terms of classification accuracy for most of the data sets. Another advantage is its computational efficiency. Even if PLS dimension reduction is originally designed for continuous regression, it can be successfully applied to classification problems. To determine the optimal number of PLS components, a new simple procedure based on random partitions is proposed. The reliability of this procedure is quite good, although not perfect. An aggregation strategy (AdaBoost) was used in the hope of im-

proving the classification accuracy, because aggregation methods are known to be very effective in reducing the error rate on independent test data. The conclusion is that boosting does not improve the classification accuracy of PLS, except in some special cases. The second topic of this paper is gene selection. We show that the first PLS component can be used for gene selection and proove that the proposed procedure is equivalent to a well-known gene selection procedure found in the literature. Thus, the information on single genes does not get lost through the PLS dimension reduction. At last, we claim that PLS can also be seen as a practical visualization tool in the context of classification. In contrary to principal component analysis, PLS is a supervised procedure which focus on class separation. Unlike sufficient dimension reduction and related methods, PLS can handle all the genes simultaneously and performs gene selection intrinsically. In a word, PLS is a very fast and competitive tool for classification problems with high-dimensional data as regards to prediction accuracy, feature selection and visualization.

Acknowledgement

I thank Gerhard Tutz, Korbinian Strimmer and Joe Whittaker for critical comments and discussion, Klaus Hechenbichler for providing the R program for AdaBoost and Jane Fridlyand for providing the pre-processed NCI data set.

References

Alizadeh, A. 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., Hudson, J., 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., Staudt, L. M., 2000. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature 403, 503–511.

Alon, U., Barkai, N., Notterman, D. A., Gish, K., Ybarra, S., Mack, D., Levine, A., 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. PNAS 96, 6745–6750.

Ben-Dor, A., Bruhn, L., Friedman, N., Nachman, I., Schummer, M., Yakhini, Z., 2000. Tissue classification with gene expression profiles. Journal of Computational Biology 7, 559–584.

Bo, T. H., Jonassen, I., 2002. New feature subset selection procedures for classification of expression profiles. Genome Biology 3, R17.

Braga-Neto, U., Hashimoto, R., Dougherty, E. R., Nguyen, D. V., Carroll, R. J., 2004. Is cross-validation better than resubstitution for ranking genes ? Bioinformatics 20, 253–258.

Breiman, L., 1996. Bagging predictors. Machine Learning 24, 123–140.

Chiaromonte, F., Martinelli, J., 2001. Dimension reduction strategies for analyzing global gene expression data with a response. Mathematical Biosciences 176, 123–144.

Culhane, A. C., Perriere, G., Considine, E., Gotter, T., Higgins, D., 2002. Between-group analysis of microarray data. Bioinformatics 18, 1600–1608.

de Jong, S., 1993. Simpls. an alternative approach to partial least squares regression. Chemometrics and Intelligent Laboratory Systems 18, 251–253.

Dettling, M., Bühlmann, P., 2003. Boosting for tumor classification with gene expression data. Bioinformatics 19, 1061–1069.

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

Dudoit, S., Shaffer, J. P., Boldrick, J. C., 2003. Multiple hypothesis testing in microarray experiments. Statistical Science 18, 71–103.

Frank, I. E., Friedman, J. H., 1993. A statistical view of some chemometrics regression tools. Technometrics 35, 109–135.

Freund, Y., 1995. Boosting a weak learning algorithm by majority. Information and Computation 121, 256–285.

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.Stat.Assoc. 89, 122–127.

Ghosh, D., 2002. Singular value decomposition regression modelling for classification of tumors from microarray experiments. Proceedings of the Pacific Symposium on Biocomputing 98, 11462–11467.

Golub, T., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P. ., Coller, H., Loh, M. L., Downing, J., Caligiuri, M. A., Bloomfield, C. D., Lander, E. S., 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286, 531–537.

Hastie, T., Tibshirani, R., Friedman, J. H., 2001. The elements of statistical learning. Springer-Verlag, New York.

Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Raffeld, M., Yakhini, Z., Ben-Dor, A., Dougherty, E., Kononen, J., Bubendorf, L., Fehrle, W., Pittaluga, S., Gruvberger, S., Loman, N., Johannsson, O., Olsson, H., Wilfond, B., Sauter, G., Kallioniemi, O.-P., Borg, A., Trent, J., 2001. Gene-expression profiles in hereditary breast cancer. N Engl J Med 344, 539–548.

Helland, I., 1988. On the structure of partial least squares. Communication in Statistics., Simulation and Computation 17, 581–607.

Huang, X., Pan, W., 2003. Linear regression and two-class classification with gene expression data. Bioinformatics 19, 2072–2078.

Kahn, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., Meltzer, P. S., 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine 7, 673–679.

Li, L., Weinberg, C. R., Darden, T. A., Pedersen, L. G., 2001. Gene selection for sample classification based on gene expression: study of sensitivity to choice of parameters of the ga/knn method. Bioinformatics 17, 1131–1142.

Martens, H., 2001. Reliable and relevant modelling of real world data: a personal account of the development of pls regression. Chemometrics and Intelligent Laboratory Systems 58, 85–95.

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

Naes, T., , Martens, H., 1985. Comparison of prediction methods for multicollinear data. Communications in Statistics, Part B – Simulation and Computation 14, 545–576.

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

Notterman, D. A., Alon, U., Sierk, A. J., Levine, A. J., 2001. Transcriptional gene expression profi les of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays. Cancer Research 61, 3124–3130.

Ross, D. T., Scherf, U., Eisen, M. B., Perou, C. M., Spellman, P., Iyer, V., Jeffrey, S. S., de Rijn, M. V., Waltham, M., Pergamenschikov, A., Lee, J. C. F., Lashkari, D., Shalon, D., Myers, T. G., Weinstein, J. N., Botstein, D., Brown, P. O., 2000. Systematic variation in gene expression patterns in human cancer cell lines. Nature Genetics 24, 227–234.

Schapire, R., Freund, Y., Bartlett, P., Lee, W., 1998. Boosting the margin: A new explanation for the effectiveness of voting methods. Annals of Statistics 26, 1651–1686.

Simon, R. M., Korn, E. L., McShane, L. M., Radmacher, M. D., Wright, G. W., Zhao, Y., 2004. Design and Analysis of DNA Microarray Investigations. Springer-Verlag, New York.

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

Stone, M., Brooks, R. J., 1990. Continuum regression: cross-validated sequentially constructed prediction embracing ordinary least squares, partial least squares and principal component regression. J.R.Statist.Soc.B 52, 237–269.

Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G., 2002. Diagnosis of multiple cancer types by shrunken centroids of gene expression. PNAS 99, 6567–6572.

Tutz, G., Hechenbichler, K., 2004. Aggregating classifi ers with ordinal response structure. Journal of Statistical Computation and Simulation to appear.

West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J., Marks, J., Nevins, J., 2001. Predicting the clinical status of human breast cancer using gene expression profi les. PNAS 98, 11462–11467.

Colon	1	2	3	4	5		m_{opt}
$(K = 2)$	0.136	0.114	0.119	0.143	0.147		0.124
Leukemia	1	2	3	4	5		m_{opt}
$(K = 2)$	0.020	0.028	0.03	0.030	0.028		0.024
Prostate	1	2	3	4	5		m_{opt}
$(K = 2)$	0.366	0.140	0.076	0.081	0.077		0.078
Breast cancer	1	2	3	4	5		m_{opt}
$(K = 2)$	0.14	0.110	0.104	0.106	0.103		0.110
Carcinoma	1	2	3	4	5		m_{opt}
$(K = 2)$	0.025	0.021	0.022	0.024	0.023		0.024
Lymphoma	1	2	3	4	5	6	m_{opt}
$(K = 3)$	0.037	0.0003	0.002	0.001	0.004	0.003	0.004
SRBCT	1	2	3	4	6	10	m_{opt}
$(K = 4)$	0.343	0.200	0.056	0.027	0.009	0.003	0.003
BRCA	1	2	3	4	5	6	m_{opt}
$(K = 3)$	0.468	0.348	0.310	0.268	0.285	0.303	0.0304
NCI	1	5	10	15	20		m_{opt}
$(K = 8)$	0.715	0.338	0.293	0.318	0.325	0.338	

Table 1: Mean error rate over 200 random partitions with PLS

Colon	55NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.182	0.19	0.134	0.139	0.143	0.130
Leukemia	55NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.034	0.039	0.038	0.05	0.022	0.046
Prostate	55NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.119	0.124	0.086	0.085	0.370	0.099
Breast cancer	55NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.117	0.123	0.100	0.093	0.120	0.147
Carcinoma	55NN-20	5NN-50	$SVM-20$	$SVM-50$	PAM	PAM-opt
$(K=2)$	0.020	0.021	0.024	0.029	0.036	0.096
Lymphoma	55NN-100	5NN-200	$SVM-100$	$SVM-200$	PAM	PAM-opt
$(K=3)$	0.014	0.003	0.038	0.019	0.013	0.042
SRBCT	55NN-100	5NN-200	$SVM-100$	$SVM-200$	PAM	PAM-opt
$(K=4)$	0.012	0.0052	0.010	0.014	0.046	0.069
BRCA	55NN-100	5NN-200	$SVM-100$	$SVM-200$	PAM	PAM-opt
$(K=3)$	0.378	0.318	0.588	0.581	0.331	0.396
NCI	55NN-100	5NN-200	$SVM-100$	$SVM-200$	PAM	PAM-opt
$(K=8)$	0.394	0.366	0.466	0.452	0.316	0.296

Table 2: Mean error rate over 200 random partitions with classical methods

	$B=1$	$B=2$	$B=3$	$B=4$	$B=5$
PLS 1	0.80	-0.74	0.79	-0.74	0.60
PLS 2	-0.48	0.63	-0.35	0.58	-0.30
PLS 3	0.03	0.00	-0.00	0.00	0.14
PLS 4	-0.06	-0.01	-0.03	-0.02	-0.19

Table 3: Correlations between 4 PLS components and the 5 first PLS components with boosting (prostate data)

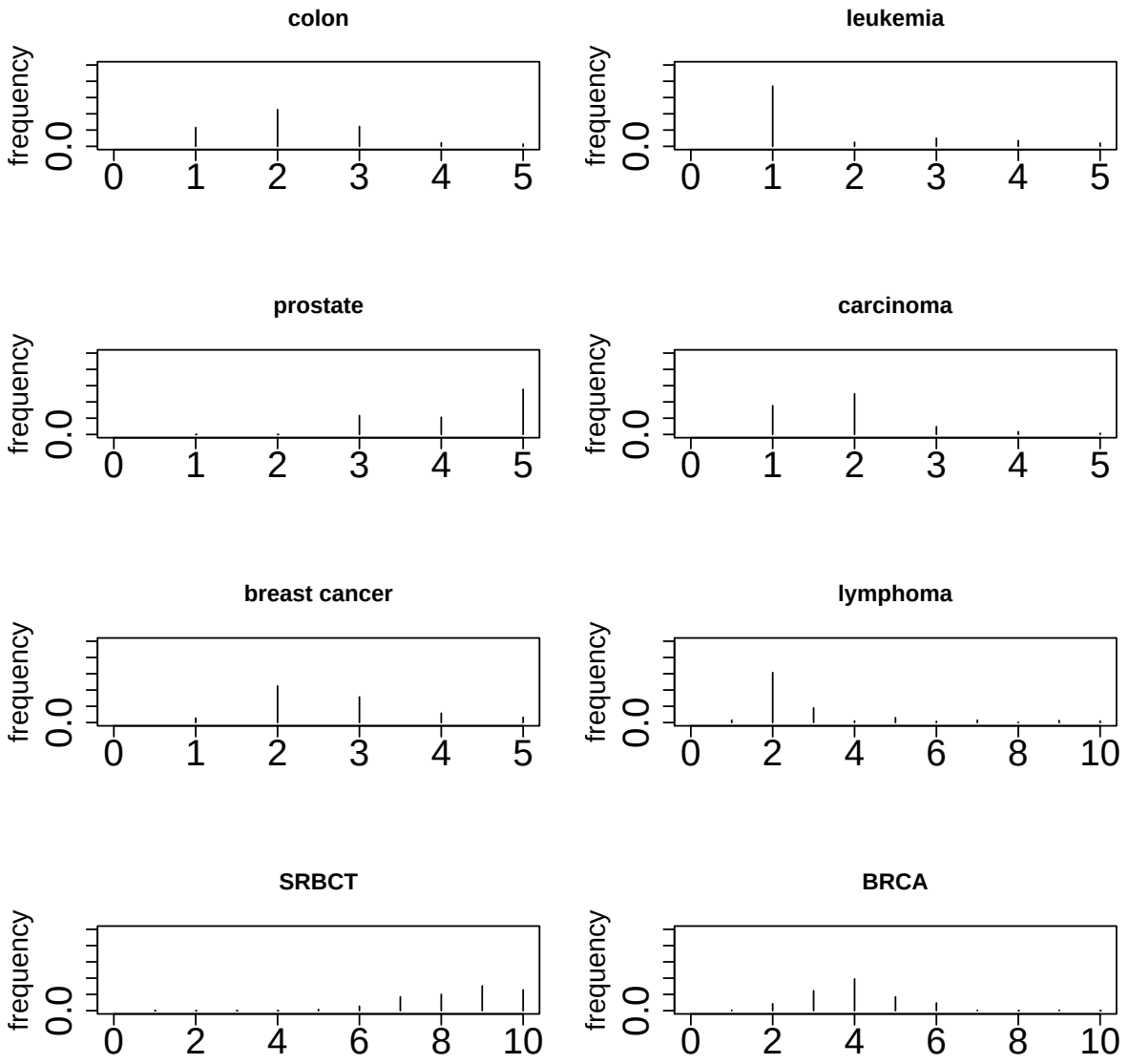

Figure 1: Histogram of the estimated optimal number of components for different data sets.

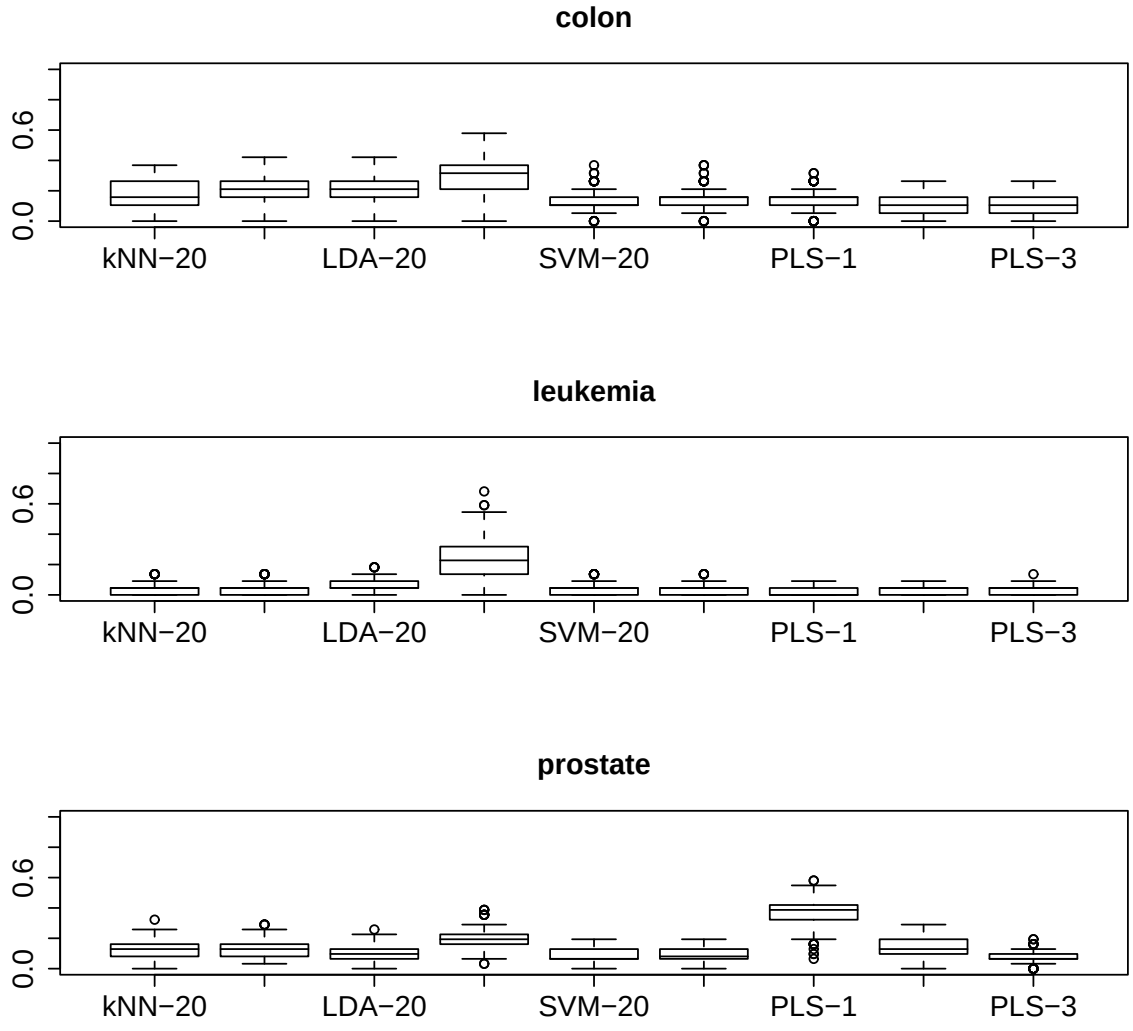

Figure 2: Boxplots of the error rate over the 200 random partitions for different classifi cation methods and different data sets

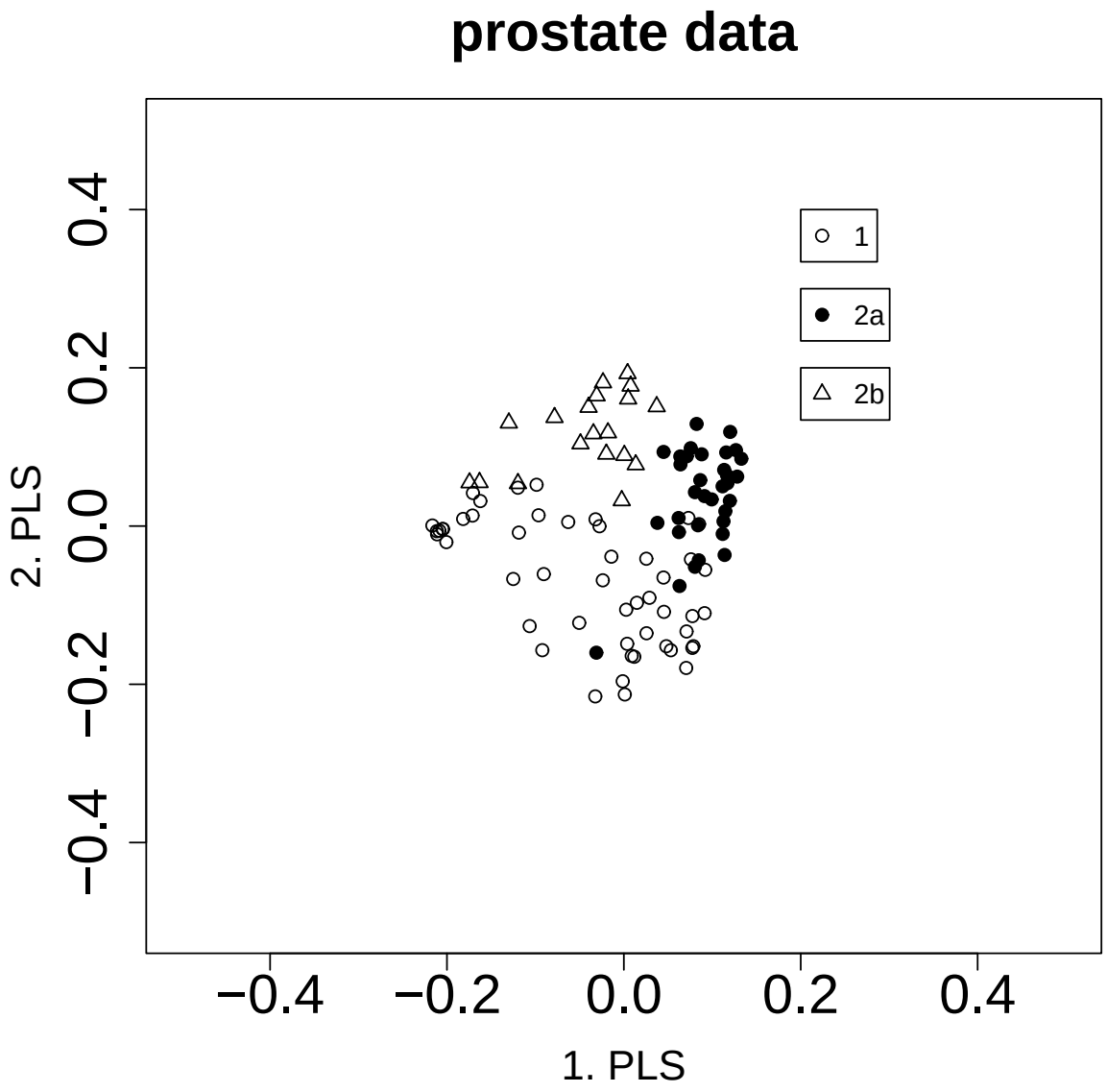

Figure 3: First and second PLS components for the lymphoma data

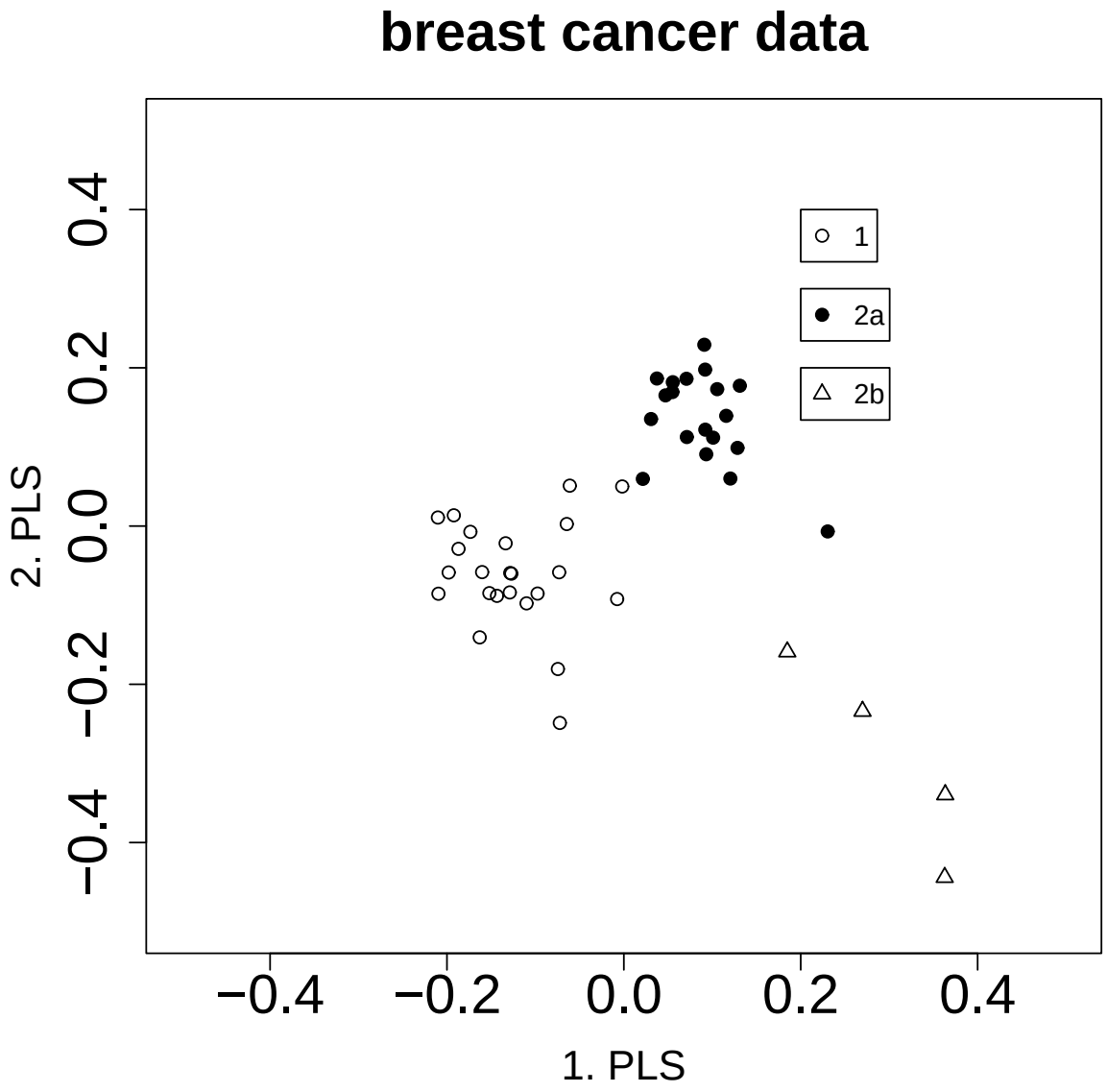

Figure 4: First and second PLS components for the breast cancer data

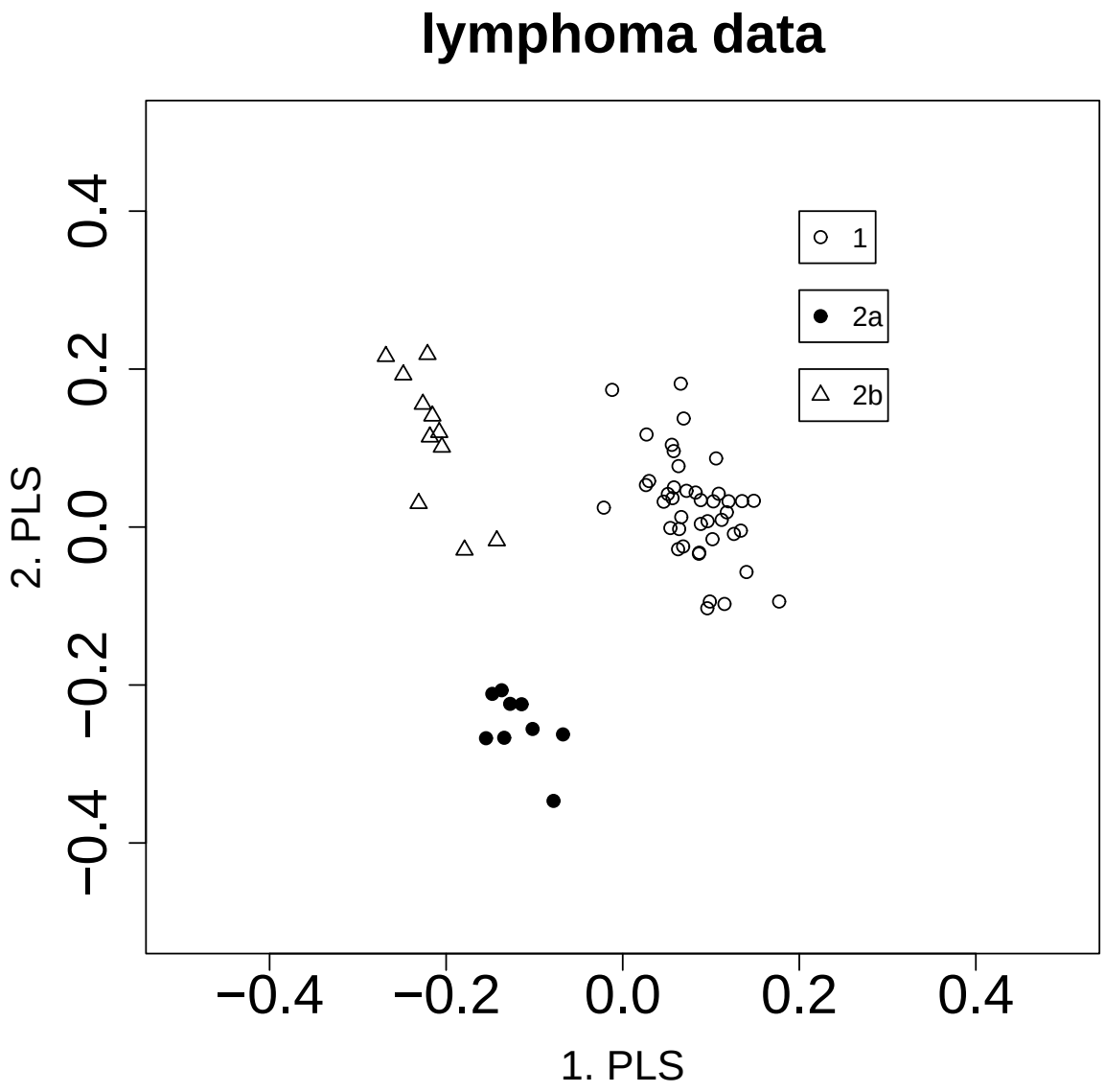

Figure 5: First and second PLS components for the lymphoma data with 2 classes