Kernel logistic PLS: a tool for supervised nonlinear dimensionality reduction and binary classification

Arthur Tenenhaus, Alain Giron, Emmanuel Viennet, Michel Béra, Gilbert Saporta, Bernard Fertil

Kernel logistic PLS: A tool for supervised nonlinear dimensionality reduction and binary classification

Arthur Tenenhaus[a, b, *], Alain Giron[a], Emmanuel Viennet[c], Michel Béra[b], Gilbert Saporta[d], Bernard Fertil[a, e]

[a] *U678 INSERM, CHU Pitié-Salpétrière, 91 bd de l'hôpital, 75634 Paris, France*
[b] *KXEN research, 25 quai Galliéni, 92150 Suresnes, France*
[c] *Laboratoire d'informatique LIPN, Université Paris XIII, France*
[d] *CNAM , 292 rue Saint Martin, case 441, 75141 Paris cedex 03, France*
[e] *Laboratoire LSIS (UMR CNRS 6168), Equipe I&M (ESIL), case 925, 163, avenue de Luminy, 13288 Marseille cedex 9, France*

Abstract

"Kernel logistic PLS" (KL-PLS) is a new tool for supervised nonlinear dimensionality reduction and binary classification. The principles of KL-PLS are based on both PLS latent variables construction and learning with kernels. The KL-PLS algorithm can be seen as a supervised dimensionality reduction (complexity control step) followed by a classification based on logistic regression. The algorithm is applied to 11 benchmark data sets for binary classification and to three medical problems. In all cases, KL-PLS proved its competitiveness with other state-of-the-art classification methods such as support vector machines. Moreover, due to successions of regressions and logistic regressions carried out on only a small number of uncorrelated variables, KL-PLS allows handling high-dimensional data. The proposed approach is simple and easy to implement. It provides an efficient complexity control by dimensionality reduction and allows the visual inspection of data segmentation.

Keywords: Classification; Kernel; PLS regression; Logistic regression; Dimensionality reduction

1. Introduction

In this paper, we revisit the classification problem by focusing on kernel-based methods such as support vector machines (SVM) (Vapnik, 1998; Burges, 1998; Boser et al., 1992), kernel logistic regression (KLR) (Zhu and Hastie, 2005) and others such as kernel partial least squares (KPLS) regression (Rosipal and Trejo, 2001; Rosipal et al., 2003; Bennett and Embrechts, 2003). In kernel-based methods, a complexity control step is usually performed. Vapnik–Chervonenkis (VC) theory provides a general framework for complexity control called structural risk minimization (SRM). VC theory relies on a nested set of models of increasing complexity (called a structure). In this context, the choice of a suitable structure is a decisive step. This paper proposes a novel way to build such a nested structure based on the partial least squares (PLS) regression. The PLS algorithm is especially convenient when handling high-dimensional and highly

* Corresponding author. U678 INSERM, CHU Pitié-Salpétrière, 91 bd de l'hôpital, 75634 Paris, France. Tel.: +33 1 53 82 84 07; fax: +33 1 53 82 84 46.

E-mail address: arthur.tenenhaus@imed.jussieu.fr (A. Tenenhaus).

correlated data (Wold, 1983; Tenenhaus, 1998), which is the very issue addressed when using kernels. This dual aspect (complexity control procedure and explicit linear model designed for high-dimensional and highly correlated data) makes the association of kernel theory with PLS regression (PLS-R) a clearly fruitful approach that has already given rise to many developments. For instance, Rosipal and Trejo (2001) have proposed a nonlinear PLS algorithm via kernel: the KPLS regression.

PLS-R was not originally designed for classification. However, based on the algorithmic structure of PLS-R, two new linear approaches have been proposed: the PLS logistic regression (Tenenhaus, 2002; Bastien et al., 2005; Tenenhaus et al., 2005) and the PLS for discrimination (Barker and Rayens, 2003). Rosipal et al. have also suggested a kernelized version of the PLS for discrimination (Rosipal et al., 2003).

Based on Bennett and Embrechts's (2003) work, we present a nonlinear extension of the PLS logistic regression based on the factorization of the kernel matrix: the Kernel Logistic PLS (KL-PLS).

The paper is organized as follows: in Section 2, we briefly review some principles of SRM and PLS-R. In Section 3, we focus on PLS for classification. In Section 4, we review some nonlinear extensions of PLS-R. In Section 5, we propose the KL-PLS algorithm and present numerical and graphical results in Section 6.

2. Partial least squares regression and structural risk minimization

2.1. Structural risk minimization (SRM)

Given a set of examples $(x_1, y_1), \ldots, (x_n, y_n)$, $x_i \in R^p$, $y_i \in R$, drawn from an unknown distribution $P(x, y)$, expected risk minimization is achieved by finding a function $f_{\text{opt}} \in F_\Lambda = \{f_\lambda, \lambda \in \Lambda\}$, which provides the smallest possible expected risk:

$$R(\lambda) = \int V(y, f_\lambda(x)) P(x, y)\, \mathrm{d}x\, \mathrm{d}y, \tag{1}$$

where each function f_λ in F_Λ is uniquely labeled by the adjustable parameter λ and V defines a cost function measuring the discrepancy between the y and $f_\lambda(x)$.

Usually, the probability distribution $P(x, y)$ is unknown, preventing minimizing expected risk. However, a sampling of $P(x, y)$ is available from which the empirical risk can be minimized:

$$R_{\text{emp}}(\lambda) = \frac{1}{n} \sum_{i=1}^{n} V(y_i, f_\lambda(x_i)). \tag{2}$$

Vapnik and Chervonenkis have provided an inequality connecting expected risk with empirical risk, with a probability of $1 - \eta$:

$$R(\lambda) \leqslant R_{\text{emp}}(\lambda) + \sqrt{\frac{h(\log(2n/h) + 1) - \log(\eta/4)}{n}} \quad \forall \lambda \in \Lambda, \tag{3}$$

where h is the VC-dimension of F_Λ (Vapnik, 1998). The right-hand side of the inequality (3) is called the "structural risk". It is clear from inequality (3) that a small value of the empirical risk does not necessarily imply a small value of the expected risk. In order to make the expected risk small, both the empirical risk and the VC-dimension should be minimized at the same time. In the context of SRM, a set of possible models $F_\Lambda = \{f_\lambda(x) : \lambda \in \Lambda\}$ is decomposed into a nested structure of subsets $F_k = \{f_\lambda(x) : \lambda \in \Lambda_k\}$ of increasing complexity (or flexibility to fit the data), so that $F_1 \subset F_2 \subset \cdots \subset F_k \subset \ldots$, and the (finite) VC-dimension h_k of F_k verifies $h_1 \leqslant h_2 \leqslant \cdots \leqslant h_k \ldots$. Thus, a model can be chosen among the nested sequence based on the smallest value of the upper bound of (3).

Application of SRM, in practice, depends on a chosen structure. An example of generic structure is a dictionary representation where the set of approximating functions is

$$f_m(x, c, w) = \sum_{i=1}^{m} c_i b(x, w_i), \tag{4}$$

where $b(x, w_i)$ is a set of basis functions with adjustable parameters w_i and c_i are linear coefficients (Cherkassky et al., 1999). Both w_i and c_i are estimated from the training data. Representation (4) defines a structure, since $f_1 \subset$

$f_2 \subset \cdots \subset f_k \subset \ldots$. Hence the number of terms m in (4) specifies a particular f_i of the structure. We formulate, in the next section, a practical way to build such a nested structure based on the PLS-R.

2.2. Partial least squares regression (PLS-R)

PLS-R (Wold et al., 1983) is a technique for summarizing two data sets X and Y by latent variables (or PLS components), taking into account that the block Y is a set of responses and the block X is a set of predictors. PLS components, denoted by $t_1, \ldots, t_m$ are related to X and constrained to be orthogonal, and PLS components, denoted by $u_1, \ldots, u_m$ are related to Y and are not constrained to be orthogonal. Let X_0 and Y_0 be the centralized versions of X and Y. Höskuldsson (1988) has shown that the hth PLS components $t_h = X_{h-1}w_h$ and $u_h = Y_{h-1}c_h$, are obtained by maximizing the Tucker (1958) criterion:

$$\mathrm{cov}^2(X_{h-1}w_h, Y_{h-1}c_h) = \mathrm{var}(X_{h-1}w_h)\, corr^2(X_{h-1}w_h, Y_{h-1}c_h)\, \mathrm{var}(Y_{h-1}c_h), \tag{5}$$

subject to the constraints $\|w_h\| = \|c_h\| = 1$. X_{h-1} is the residual of the regression of X_0 on $t_1, \ldots, t_{h-1}$ and Y_{h-1} is the residual of the regression of Y_0 on $t_1, \ldots, t_{h-1}$. The solution w_h (respectively, c_h) of this problem is obtained from the normalized eigenvector of the matrix $X_{h-1}^t Y_{h-1} Y_{h-1}^t X_{h-1}$ (respectively, $Y_{h-1}^t X_{h-1} X_{h-1}^t Y_{h-1}$) associated with the largest eigenvalue.

Therefore, PLS creates orthogonal latent variables $t_1, t_2, \ldots, t_m$ (PLS components), linear combinations of the p original variables $x_1, \ldots, x_p$ using Y for their determination. Let us focus on the one-dimensional output case. The PLS model with m components fits expansion (4) as follows:

$$f_m(x, c, w) = \sum_{i=1}^{m} c_i b(x, w_i) = \sum_{h=1}^{m} c_h t_h = \sum_{h=1}^{m} c_h \sum_{j=1}^{p} x_{h-1,j} w_{hj} = \sum_{h=1}^{m} c_h \sum_{j=1}^{p} x_j w_{hj}^*, \tag{6}$$

where $w_h = \underset{w_h}{\arg\max}\ \mathrm{cov}(X_{h-1}w_h, y)$, c_h is the regression coefficient of t_h in the regression of y on $t_1, \ldots, t_m$ and w_h^* is defined in Section 5.5. It is worth noting that y is integrated in the construction of the nested structure. Subsequently, the nested structure of f_i is built in a supervised way and complexity control is achieved here by supervised dimensionality reduction.

3. Linear PLS methods for classification

Although PLS was not inherently designed for classification, it is routinely used for this purpose. PLS discriminant analysis (PLS-DA) is usually defined as a PLS-R of the dummy matrix derived from y on the set of predictors X. Thus PLS-DA is a technique for modeling linear relation between a set of binary variables Y on the set of predictors X. In addition, two linear PLS methods have been designed especially for classification: PLS for discrimination and PLS logistic regression.

3.1. PLS for discrimination (PLS-D)

In this section, we assume that Y is the dummy matrix. Barker and Rayens (2003) suggested to remove the not meaningful Y-space penalty, $\mathrm{var}(Y_{h-1}c_h)$, in criterion (5). They have noted that this penalty is, in fact not adequate for classification and proposed instead to obtain the PLS-D components $t_h = X_{h-1}w_h$ and $u_h = Yc_h$, $h = 1, \ldots, m$, by maximizing criterion:

$$\mathrm{var}(X_{h-1}w_h)\, corr^2(X_{h-1}w_h, Yc_h), \tag{7}$$

subject to the constraints $\|w_h\| = \|c_h\| = 1$. X_{h-1} is the residual of the regression of X_0 on $t_1, \ldots, t_{h-1}$. The solution w_h is obtained from the normalized eigenvector of the matrix $X_{h-1}^t Y (Y^t Y)^{-1} Y^t X_{h-1}$ associated with the largest eigenvalue. We may note that $t_h = X_{h-1}w_h$ is the first principal component of the PCA of the X_{h-1} variables projected onto the Y variable space.

As far as one-dimensional output is concerned, the PLS components extracted from PLS-D and PLS-DA are identical. This follows from the fact that $Y^t Y$ is a scalar.

PLS-DA and PLS-D only extract PLS components and a supervised method (such as the logistic regression) applied on selected components, provides a classification rule.

3.2. PLS logistic regression (PLS-LR)

When the set of responses $Y\,(=y)$ is limited to one variable, the algorithm can be simplified because the ranks of the matrices $X^t yy^t X$ and $y^t XX^t y$ are equal to one. The following results are obtained:

$$w_1 = X^t y/\|X^t y\|,$$

$$t_1 = Xw_1 = XX^t y/\|X^t y\|.$$

Garthwaite (1994) has noticed that each coordinate of the weight vector w_1 can be written as

$$w_{1j} = x_j^t y/\|X^t y\| = x_j^t x_j \frac{x_j^t y}{x_j^t x_j \cdot \|X^t y\|} \propto s_j^2 a_j \Big/ \sqrt{\sum_j s_j^2 a_j},$$

where s_j^2 is the variance of x_j and a_j the regression coefficient of x_j in the regression of y on x_j. Therefore, the weight vector w_1 is obtained by a succession of simple regressions of the dependent variable y on each explanatory variable x_j. This result remains valid for the next components (w_h is obtained by a succession of multiple regressions of y on $t_1, \ldots, t_{h-1}$ and each residual variable $x_{h-1,j}$).

Using the Garthwaite (1994) approach and replacing the successions of simple regressions by simple generalized linear regressions, Bastien et al. (2005) extended the PLS-R to the PLS generalized linear regression (PLS-GLR). The PLS-GLR algorithm provides a natural tool for classification through the logistic regression (since binary logistic regression belongs to the generalized linear model family) giving rise to the PLS logistic regression (PLS-LR) (Tenenhaus, 2002, 2005; Bastien et al., 2005). The PLS-GLR algorithm is described in Appendix A.

The PLS-GLR of a response y on a set of predictors $x_1, \ldots, x_p$ with m components is written as

$$g(\theta) = \sum_{i=1}^{m} c_h \left(\sum_{j=1}^{p} w_{hj}^* x_j \right), \tag{8}$$

where the parameter θ is the mean of a numerical variable. The link function, g, is chosen by the user according to the probability distribution of y.

Below are quoted four important properties of PLS-LR:

(i) Contrary to many data analysis methods, it needs only to compute the inverse of small-dimensional matrices (size equal to the number of PLS-LR components). The number of components is generally small and usually chosen by cross-validation.
(ii) It allows handling data with more variables than observations.
(iii) It allows handling highly correlated variables.
(iv) Eqs. (8) and (4) have a similar expansion so that PLS-LR (as well as PLS-R) fits the SRM paradigm.

PLS-DA, PLS-D and PLS-LR are designed to operate with high-dimensional and highly correlated data. The ability of these methods to do inference in high-dimensional space makes them ideal candidates for a nonlinear analysis of data based on an increase of their dimension.

4. Nonlinear PLS methods

Several methods are devoted to the extension of the linear PLS methods to the nonlinear case. Focusing here on binary classification we briefly review pros and cons of some of them, in order to precisely substantiate our contribution.

4.1. The simplest approach

One simple way of introducing nonlinearity in the model is to increase the dimension of the input matrix X by using higher order and cross terms of the original input variables, and then to use this new matrix X in the PLS algorithm (Wold et al., 1989). However, such an increase is workable only if p is originally very small. It is worth noting that the interpretability of the final model (coefficients and variables) is preserved.

4.2. The Wold's approach

Without discarding PLS as a tool for dimensionality reduction, the linear inner relation

$$\hat{y} = \sum_{h=1}^{m} c_h \sum_{j=1}^{p} w_{hj}^{*} x_j = \sum_{h=1}^{m} c_h t_h, \tag{9}$$

may be replaced with a nonlinear relation yielding the following general form of $\hat{y}$:

$$\hat{y} = \sum_{h=1}^{m} g_h \left(\sum_{j=1}^{p} w_{hj}^{*} x_j \right) = \sum_{h=1}^{m} g_h(t_h), \tag{10}$$

where g_h's are smooth nonlinear functions. This was first proposed by Wold using quadratic polynomials without cross terms (Wold et al., 1989) and further developed with smooth functions approximated by splines (Wold, 1992) or neural networks (Baffi et al., 1999).

The initial PLS dimension reduction is particularly useful when a set of variables with no discriminant information jeopardizes data segmentation. However, if the discriminant information relies on nonlinear relationships between variables and target, the initial PLS dimension reduction may have a disastrous effect.

4.3. KPLS approach

Kernel-based learning algorithms (Cortes and Vapnik, 1995; Schölkopf and Smola, 2002; Shawe-Taylor and Cristianini, 2004) proceed by mapping original data X into a (possibly high-dimensional) Hilbert space $\mathtt{F}$, usually called feature space, and then by using linear pattern analysis to detect relations in the feature space. Mapping is performed by specifying the inner product between each pair of data rather than by explicitly calculating their coordinates (the so-called "kernel trick"). This approach has several advantages, the most important being that the inner product in the feature space is often much more easily computed than the coordinates of the points (notably when the dimensionality of the feature space is high). Given an input space $\mathtt{X}$ and a feature space F, we consider a function $\Psi\colon \mathtt{X} \to \mathtt{F}$. Given two points $u \in \mathtt{X}$ and $v \in \mathtt{X}$, the kernel function returns the inner product between their images mapped by Ψ. The choice of the kernel function (and associated parameter(s)) defines the relative position of the data in the feature space. The linear kernel ($k(u, v) = \langle u, v \rangle$) and the Gaussian kernel ($k(u, v) = \exp(-\|u - v\|^2/2\sigma^2)$) are two of the most popular kernel functions.

The nonlinear KPLS regression method is based on the mapping of the original input data X into a high-dimensional feature space $\mathtt{F}$. The objective is to construct a linear PLS-R in $\mathtt{F}$ or equivalently a nonlinear PLS-R model in $\mathtt{X}$. Rosipal and Trejo developed a nonlinear form of the PLS-R based on the kernel trick, the so-called KPLS regression (Rosipal and Trejo, 2001). In the case of one-dimensional output, the first PLS component t_1^{PLS} is proportional to $XX^t y$ and thus, PLS can be rigorously adapted to the kernel approach, giving rise to the KPLS.

Denote Φ the matrix of mapped input data into the feature space $\mathtt{F}$.

$$t_1^{\mathrm{KPLS}} \propto \Phi\Phi^t y. \tag{11}$$

Instead of an explicit nonlinear mapping, the kernel trick can be used.

$$t_1^{\mathrm{KPLS}} \propto \Phi\Phi^t y = Ky, \tag{12}$$

where K is the $n \times n$ kernel matrix (see Section 5.1).

In Section 2.2, we assumed a centralized PLS-R problem. To centralize the mapped data in a feature space F we can apply (13) (Schölkopf et al., 1998; Rosipal and Trejo, 2001):

$$K_0 = \left(I_n - \frac{1}{n} 1_n 1_n^t\right) K \left(I_n - \frac{1}{n} 1_n 1_n^t\right), \tag{13}$$

where I_n is an $n \times n$ identity matrix and 1_n represents a $n \times 1$ vector of ones.

Like PLS, KPLS is an iterative process. After extraction of the unit norm t_{h-1}^{KPLS}, the $(h-1)$th KPLS component, the algorithm is reiterated using the following deflated matrix:

$$h \geqslant 2, \quad K_{h-1} = (I - t_{h-1}^{\mathrm{KPLS}} t_{h-1}^{\mathrm{KPLS}^t}) K_{h-2} (I - t_{h-1}^{\mathrm{KPLS}} t_{h-1}^{\mathrm{KPLS}^t}). \tag{14}$$

Eq. (14) is based on the fact that Φ_{h-1} is deflated as

$$h \geqslant 2, \quad \Phi_{h-1} = \Phi_{h-2} - t_{h-1}^{\mathrm{KPLS}} t_{h-1}^{\mathrm{KPLS}^t} \Phi_{h-2}, \tag{15}$$

and the hth unit norm KPLS component is defined by

$$t_h^{\mathrm{KPLS}} = K_{h-1} y. \tag{16}$$

Like PLS, KPLS is not designed for classification purposes. Therefore, Rosipal et al. (2003) have suggested a kernelized version of the PLS for discrimination (KPLS-D). As it is for the linear framework, KPLS-D and KPLS-DA for binary classification are equivalent. However, KPLS-D provides an appealing solution for the multiclass classification (Rosipal et al., 2003) and should be preferred to KPLS-DA.

It is worth pointing out that KPLS allows avoiding drawbacks of the two first nonlinear PLS methods described above. Indeed, PLS component in the feature space allows capturing nonlinear relation between X and y. Moreover, it is "only" necessary to handle a $n \times n$ kernel matrix and KPLS limitations are much more related to the number of observations (n) than to the number of variables (p). However, the interpretability of the model is lost.

4.4. Direct kernel PLS (DK-PLS)

Bennett and Embrechts (2003) introduced the DK-PLS regression. They suggested applying PLS-R of y on K_0 in order to obtain a low rank approximation of the kernel matrix (suitable for regression), used to obtain the final linear model. DK-PLS setting is based on the so-called "empirical kernel map" (also called "direct kernel mapping") (Schölkopf and Smola, 2002; Webb, 1996). Columns of the kernel matrix are the new input variables. It is worth noting that DK-PLS is a particular case of the first approach (Section 4.1). In addition, when using the kernel matrix, DK-PLS allows avoiding drawbacks of the two first approaches (cf. Sections 4.1 and 4.2). However, limitation of the third approach (cf. Section 4.3) related to the number of observations is not avoided, although combining empirical kernel map with sampling of the columns of K may help escaping it. We highlight here the flexibility of the empirical kernel map. Finally, like KPLS, interpretability of the model is lost.

In the case of a one-dimensional output, the first DK-PLS component t_1^{DKPLS}, is proportional to $K_0 K_0^t y$ and K_0 being symmetric:

$$t_1^{\mathrm{DKPLS}} = K_0^2 y. \tag{17}$$

In a similar way, the hth DKPLS component is defined by

$$t_h^{\mathrm{DKPLS}} = K_{h-1}^2 y, \tag{18}$$

where K_{h-1} is the deflated matrix obtained by the following procedure:

$$K_h = K_{h-1} - t_h^{\mathrm{DKPLS}} t_h^{\mathrm{DKPLS}^t} K_{h-1}. \tag{19}$$

KPLS and DK-PLS are strongly related. In fact, for one-dimensional output response, it is easy to show that the PLS-R of y on Φ (KPLS) is the PLS-R of y on $K^{1/2}$ (DK-PLS).

5. Kernel logistic PLS (KL-PLS)

Noting that KPLS and DKPLS are competitive with other kernel-based classifiers, we propose a nonlinear classification algorithm that combines the flexibility of the empirical kernel map with the supervised properties of the PLS logistic regression: the KL-PLS. KL-PLS can be viewed as a low-rank approximation of the kernel matrix oriented for classification. The main idea of KL-PLS is to look for a discriminant space spanned by the KL-PLS components $(t_1, \ldots, t_m)$, where a simple model, such as the logistic regression may become efficient for classification. We use the kernel matrix to map the data into a higher-dimensional space where the probability of finding the hyperplane increases with the dimension of the space.

Therefore, KL-PLS computes orthogonal latent variables in the space induced by the kernel matrix before performing logistic regression in the derived feature space.

KL-PLS is consequently a 3-step algorithm:

Step 1: Computation of the kernel matrix.

Step 2: Computation of the KL-PLS components.

Step 3: Logistic regression of Y on the m retained KL-PLS components.

5.1. Construction of the kernel matrix

Let X be an $n \times p$ matrix representing n observations on p explanatory variables x_k, $k = 1, \ldots, p$ and $y \in \{0, 1\}$ a binary variable (the target) observed on the n units. Let K be a kernel matrix associated with X. K may be generated by the kernel function given in Section 4.3. The dimension of the kernel matrix is $n \times n$. Each cell k_{ij} is an inner product between the individuals i and j (holds a measure of similarity) in the feature space $\mathtt{F}$. Each column k_j, $j = 1, \ldots, n$ of K, measures similarities between individual j and the whole data set.

5.2. Construction of the first KL-PLS component

The first KL-PLS component t_1 provides the first discriminant axis. This component is built by performing a succession of n simple logistic regressions:

Step 1: Computation of the coefficients a_{1j} of k_j in the binary logistic regression of y on each k_j, $j = 1, \ldots, n$.

Step 2: Normalization of the column vector a_1 made by a_{1j}'s: $w_1 = a_1/\|a_1\|$.

Step 3: Computation of the first KL-PLS component as $t_1 = K w_1$.

Consequently, there is no matrix inversion during the computation of the first KL-PLS component.

5.3. Deflation procedure

Let us assume that the KL-PLS components $t_1, \ldots, t_{h-1}$ are already computed. This step is designed to obtain components orthogonal to $t_1, \ldots, t_{h-1}$, holding residual information on y. The hth KL-PLS component is subsequently computed from the residual of the regression of each k_j, $j = 1, \ldots, n$ on $t_1, \ldots, t_{h-1}$.

Step 1: Computation of the residual $k_{h-1,1}, \ldots, k_{h-1,n}$ from the multiple regression of each k_j, $j = 1, \ldots, n$ on $t_1, \ldots, t_{h-1}$. Let $K_{h-1} = [k_{h-1,1}, \ldots, k_{h-1,n}]$ be the residual matrix.

The residual matrix (or the deflated matrix) is performed by computing a succession of n regressions on a small number of uncorrelated variables.

5.4. Computation of the hth KL-PLS component

The hth KL-PLS component has to capture the discriminant information not available in the $h-1$ previous ones. We compute the hth KL-PLS component by performing n logistic regressions on h uncorrelated variables.

Step 1: Computation of the coefficients a_{hj} of $k_{h-1,j}$ in the binary logistic regression of y on $t_1, \ldots, t_{h-1}$ and each $k_{h-1,j}$, $j = 1, \ldots, n$.

Step 2: Normalization of the column vector a_h made by a_{hj}'s: $w_h = a_h/\|a_h\|$.
Step 3: Computation of the hth KL-PLS component as $t_h = K_{h-1}w_h$.
Step 4: Expression of the component t_h in terms of K as $t_h = Kw_h^*$.
Consequently, during the construction of the hth KL-PLS component, we only compute the inverse of n h-dimensional matrices.

5.5. Expression of KL-PLS components in terms of original variables

Expression of KL-PLS components, in terms of original variables, is a fundamental step to analyze new data. Let *Ktest* be the new data set. The product $Ttest = Ktest \times W^*$ allows to compute the values of the KL-PLS components for the new data set.
Computation of w_h^*:

(a) The first KL-PLS component is already expressed in terms of original variables: $t_1 = Kw_1$ and $w_1^* = w_1$.
(b) The second KL-PLS component is expressed in terms of residuals in the regression of the original variables on t_1. Let p_{1j} be the regression coefficient of t_1 in the regression of k_j on t_1. From $K = t_1 p_1^t + K_1$ and $t_2 = K_1 w_2$ we get

$$t_2 = K_1 w_2 = (K - t_1 p_1^t)w_2 = (K - Kw_1 p_1^t)w_2 = K(I - w_1 p_1^t)w_2 = Kw_2^*.$$

(c) In a similar way, it can be shown that t_h is expressed in terms of the original variables as

$$t_h = K_{h-1}w_h = \left(K - \sum_{i=1}^{h-1} t_i p_i^t\right) w_h = \left(K - \sum_{i=1}^{h-1} Kw_i^* p_i^t\right) w_h = K\left(I - \sum_{i=1}^{h-1} w_i^* p_i^t\right) w_h = Kw_h^*.$$

5.6. Final model

We perform a binary logistic regression of y on the m retained KL-PLS components:

$$P(y = 1/X = x) = \frac{e^{\alpha_0 + \sum_{h=1}^{m} \alpha_h t_h}}{1 + e^{\alpha_0 + \sum_{h=1}^{m} \alpha_h t_h}}, \tag{20}$$

where the choice of the dimension m is guided by model selection techniques such as cross validation[1].
As mentioned in Section 5.5, the KL-PLS components $t_1, \ldots, t_h$ are linear combinations of the columns k_j of K. Thus, Eq. (20) can also be rewritten as

$$P(y = 1/X = x) = \frac{e^{\alpha_0 + \sum_{h=1}^{m} \alpha_h t_h}}{1 + e^{\alpha_0 + \sum_{h=1}^{m} \alpha_h t_h}} = \frac{e^{\alpha_0 + \sum_{j=1}^{n} \beta_j k_j}}{1 + e^{\alpha_0 + \sum_{j=1}^{n} \beta_j k_j}}. \tag{21}$$

Consequently, in order to compute $P(y = 1/X = x_{\text{new}})$, for a new observation x_{new}, it is worth noting that there is no need to compute the coordinates of x_{new} in the space spanned by the KL-PLS components.

5.7. Kernel storage properties

Strategies for improving kernel methods through factorization have been receiving increasing attention (Bach and Jordan, 2005; Smola and Schölkopf, 2000; Williams and Seeger, 2000). In fact, for example, KPLS requires storing in memory the full kernel matrix, which is a limiting factor for large training data set. Factorization appears as a very

[1] According to Bastien et al. (2005) the computation of the PLS-LR component t_h may be simplified by setting non-significant regression coefficients a_{hj} to 0 according to the Wald test. Consequently, only variables that are significantly related to y, contribute to the computation of t_h. So, the number m of KL-PLS components to be retained may be chosen by observing that the component t_{m+1} is not significant because none of the coefficients $a_{m+1,j}$ is significantly different from 0.

fruitful line of research. Below are some facts that impact kernel storage for KL-PLS:

(i) During the construction of the latent variables, KL-PLS only requires an individual column of the kernel matrix.
(ii) As suggested by Bennett and Embrechts, the introduction of an intercept when constructing the latent variables (step 1) avoids the kernel centering method proposed by Wu et al. (1997).
(iii) The succession of simple regressions used within KL-PLS algorithm provides an efficient deflation of the kernel matrix.

Consequently, it is not necessary to store the entire kernel matrix in memory but only columns one at a time. Such an implementation, however, significantly slows down the algorithm because it is necessary to reevaluate the Gram Matrix for each KL-PLS component. However, the KL-PLS components may be calculated from a subset of variables resulting from a sampling of the columns of K when the analysis of large dataset is of concern. Sampling scheme could be based on the Wald test or the likelihood ratio test: only variables that are significantly related to y in the simple logistic regression of y on each k_j, $j = 1, \ldots, n$, contribute to the construction of K. This approach is not explored in the current paper.

5.8. Complete separation

Let us investigate three data configurations: complete separation, quasi-complete separation and overlap. According to Allison (1999), there is a complete separation of data points if a vector w such as

$$\begin{cases} w^t x_j > 0 & \text{for } y_j = 1 \\ w^t x_j < 0 & \text{for } y_j = 0 \end{cases}$$

exists and quasi-complete separation of data points if a vector w such as

$$\begin{cases} w^t x_j \geqslant 0 & \text{for } y_j = 1 \\ w^t x_j \leqslant 0 & \text{for } y_j = 0 \end{cases}$$

exists. The maximum likelihood estimate exists only in the overlap configuration. Therefore, when applying logistic regression on the KL-PLS components, it is important to care about the data structure. For a complete or quasi-complete separation, Linear Discriminant Analysis or the Heinze and Schemper approach (Heinze and Schemper, 2002) consisting of reducing the bias of maximum likelihood using Firth (1993) technique, could be used. The reader is referred to Allison (1999) and Albert and Anderson (1984) for details on problems arising from (quasi-) complete data separation in the logistic regression framework.

6. Validation

In this section, we present experimental results for 11 benchmark data sets and three medical classification problems.

6.1. Benchmarks

KL-PLS efficiency is tested on 11 benchmark data sets (two-class discrimination) used in Rätsch et al. (2001) and available at http://ida.first.fraunhofer.de/projects/bench/benchmarks.htm. Details can be found in Rätsch et al. (2001). Each data set consists of 100 different training and testing partitions. Dimension of data, training sample size and test sample size of each benchmark are presented in Table 1.

Different methods have been performed for each data set and each partition. Mean and standard deviation of the test set error rate of each method have been computed using results of the 100 runs (for KLR, Zhu and Hastie computed mean and standard deviation using only 20 partitions). These values are reported in Table 2. Rätsch et al. (2001) performed SVMs. Rosipal et al. (2003) evaluated Kernel PLS-SVC (KPLS-SVC). KPLS-SVC is based on the KPLS for discrimination as dimensionality reduction method followed by SVM on retained KPLS components for the final classification. Zhu and Hastie (2005) proposed the KLR.

Table 1
Basic information about the benchmark data sets used in the study

Data set	Dimension	Training sample size	Test sample size
Banana	2	400	4600
B. cancer	9	200	77
Diabetes	8	468	300
German	20	700	300
Heart	13	170	100
Ringnorm	20	400	7000
F. solar	9	666	400
Thyroid	5	140	75
Titanic	3	150	2051
Twonorm	20	400	7000
Waveform	21	400	4600

Table 2
Comparison of classification errors (mean $\pm$ standard deviation) (test sets) for SVM (Rätsch et al., 2001), KPLS-SVC (Rosipal et al., 2003), KLR (Zhu and Hastie, 2005) and KL-PLS

Data set	SVM	KLR	KPLS-SVC	KL-PLS and parameters
Banana	$11.5 \pm 0.5^*$	$10.3 \pm 0.5^*$	$10.5 \pm 0.4^*$	10.7 ± 0.5 (0.9, 10)
B. cancer	26.0 ± 4.7	25.9 ± 4.8	$25.1 \pm 4.5^*$	25.8 ± 4.4 (50, 7)
Diabetes	$23.5 \pm 1.7^*$	?	23.0 ± 1.7	23.0 ± 1.7 (60, 4)
German	23.6 ± 2.1	23.5 ± 2.5	23.5 ± 1.6	23.2 ± 2.1 (20, 3)
Heart	16.0 ± 3.3	15.8 ± 3.5	16.5 ± 3.6	16.1 ± 3.2 (200, 2)
Ringnorm	$1.66 \pm 0.12^*$	$1.97 \pm 0.29^*$	1.43 ± 0.10	1.44 ± 0.09 (12, 1)
F. solar	$32.4 \pm 1.8^*$	33.7 ± 1.6	$32.4 \pm 1.8^*$	32.7 ± 1.8 (20, 2)
Thyroid	4.80 ± 2.19	5.00 ± 3.02	4.39 ± 2.1	4.36 ± 1.99 (15, 6)
Titanic	22.4 ± 1.0	22.4 ± 1.0	22.4 ± 1.1	22.4 ± 0.4 (300, 2)
Twonorm	$2.96 \pm 0.23^*$	2.45 ± 0.15	$2.34 \pm 0.11^*$	2.37 ± 0.10 (40, 1)
Waveform	$9.88 \pm 0.43^*$	$10.13 \pm 0.47^*$	$9.58 \pm 0.36^*$	9.74 ± 0.46 (13, 4)
Mean rank	3.1	2.9	1.9	1.9

The last column provides the width of the Gaussian kernel and the number of retained KL-PLS components. For each data set, the methods were ordered from rank 1 to rank 4 according to their performances; the last row of the table provides the mean rank for SVM, KLR, KPLS-SVC and KL-PLS. The star indicates that results of the concerned method and KL-PLS are statistically different ($p < 0.05$) (pairwise T-test for all comparisons except for KLR (two-sample T-test)).

The methods presented above are compared to KL-PLS. Gaussian kernel is used throughout this study because of its flexibility. Thus, KL-PLS efficiency relies on the value of width of the Gaussian and the number of retained KL-PLS components. These values have been selected based on the minimum classification error observed after cross validation on the first five training sets.

A pairwise T-test is used to compare KL-PLS to the other classifiers for each data set, except for KLR. A less powerful two-sample T-test is used instead, since individual classification errors are not available for that classifier. The significance level for all tests has been set to $\alpha = 0.05$.

KL-PLS vs. SVM: The classification error rate of KL-PLS is lower for 10 data sets out of 11. The null hypothesis is rejected for five of these data sets (banana, diabetes, ringnorm, twonorm, and waveform). Conversely, the classification errors rate of SVM is lower for one data set (flare-solar) and the null hypothesis is rejected in this case.

KL-PLS vs. KPLS-SVC: The classification error rate of KL-PLS is lower for four data sets out of 11. However, the null hypothesis is never rejected. Conversely, the classification errors rate of KPLS-SVC is lower for six data sets and the null hypothesis is rejected for banana, breast-cancer, flare-solar, twonorm, waveform.

The good behavior of KPLS-SVC may be related to the use of the linear SVM model, which provides a boundary following the maximal margin principle. However, the quality of results relies on a larger set of free parameters (one additional parameter is required for the SVM step), which may be bothersome in practice.

KL-PLS vs. KLR: The classification error rate of KL-PLS is lower for eight data sets out of 10 and the null hypothesis is rejected for Ringnorm and Waveform. Conversely, the classification errors rate of KLR is lower for two data sets and the null hypothesis is rejected for Banana.

Ranks of performances over all data sets offer another way to compare models (Table 2). It seems that models fall into 2 groups for which the complexity control differs: KPLS-SVC and KL-PLS—which are supervised dimensionality reduction-based models—most often provide the best results as compared to SVM and KLR for which the complexity control is achieved by Tikhonov regularization (Tikhonov and Arsenin, 1977; Evgeniou et al., 1999; Wahba, 1999).

It is worth noting that both KLR and KL-PLS are based on the logistic regression model and that the main difference concerns the complexity control scheme. To what extent difference in performances relies on the complexity control method remains to be studied.

From these experiments, we can conclude that KL-PLS works at least as well as the other methods. It is remarkable that best methods achieved very comparable results on the benchmark data sets. In fact, there may be an irreducible level of error (linked to the data structure) already reached by some of them.

To conclude, the enlarged feature space generated by the use of kernel-based methods gives flexibility, and complexity control allows good generalization. Results presented on these 11 benchmark datasets highlight the interest of the supervised dimensionality reduction approach.

6.2. Banana data set

Banana is an artificial 2-dimensional data set with two classes (400 training set, 4600 testing set) allowing a visual inspection of the data separation in the original space. According to the 2-dimensional representation, it is essentially a nonlinear classification problem. Subsequently, it seems interesting to examine classification results in more details.

6.2.1. Banana data projection onto the two first components found by KL-PLS

Fig. 1 depicts projection of the original training (left) and testing data (right) onto the two first KL-PLS components (training data).

A nice linear separation of the two classes can be observed in the derived feature space showing that a linear logistic regression is adequate to achieve an efficient classification.

6.2.2. KL-PLS and probability

KL-PLS offers a natural estimate of the conditional probability, $p(x) = P(y = 1/X = x)$ by using logistic regression as final function. Fig. 2 depicts banana data onto the original space (left) and the two first KL-PLS components (right). Colors symbolize conditional probability values $p(x)$ and lines symbolize isoprobability contours.

Linear decision boundaries in the space induced by the KL-PLS components correspond to complex nonlinear boundaries in the original space.

6.2.3. Banana test error as a function of the number of KL-PLS components

Fig. 3 shows performances of the KL-PLS model as function of the number of retained KL-PLS components. For banana data set, we note that even when we consider an important number of KL-PLS components, there is no overfitting. Moreover, the tuning of KL-PLS is quite easy. In addition to the kernel parameters, there is only one intuitive parameter (with a small number of discrete values) to adjust as compared to the continuous parameter (C) for SVM for example.

6.2.4. Kernel logistic PLS and robustness

Robustness is defined here as the quality of the model to tolerate noise. In this experiment, the training variable y is degraded by flipping the labels of randomly selected observations while preserving class balance. The test set

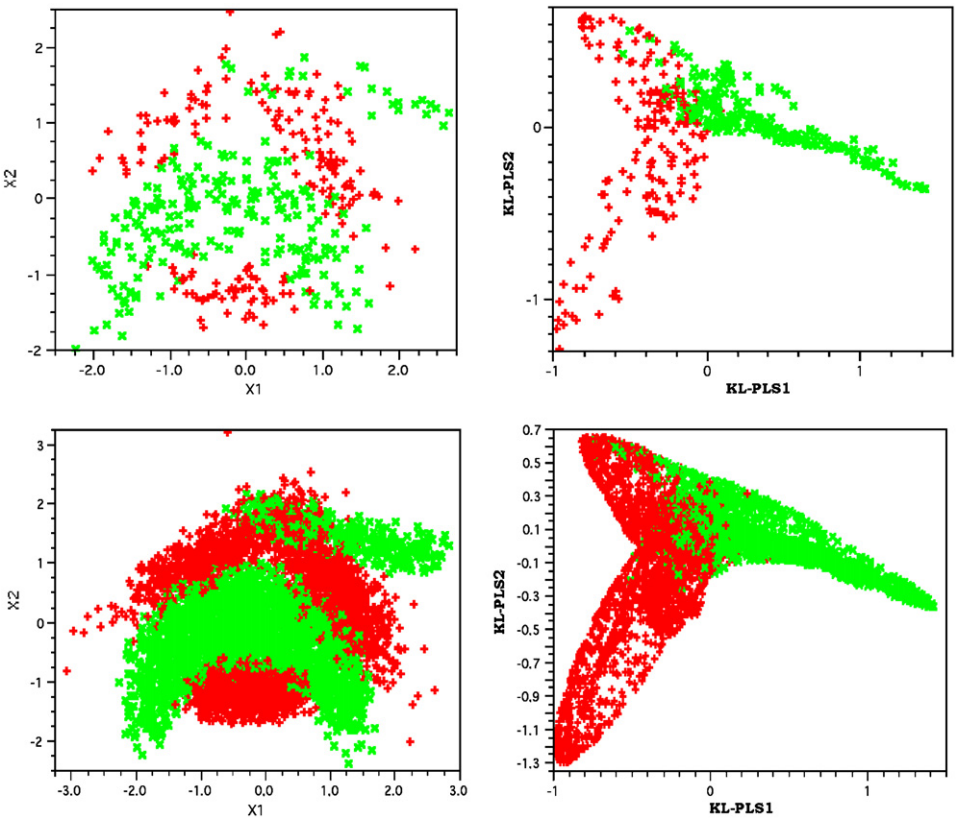

Fig. 1. Banana data depicted onto the two first KL-PLS components.

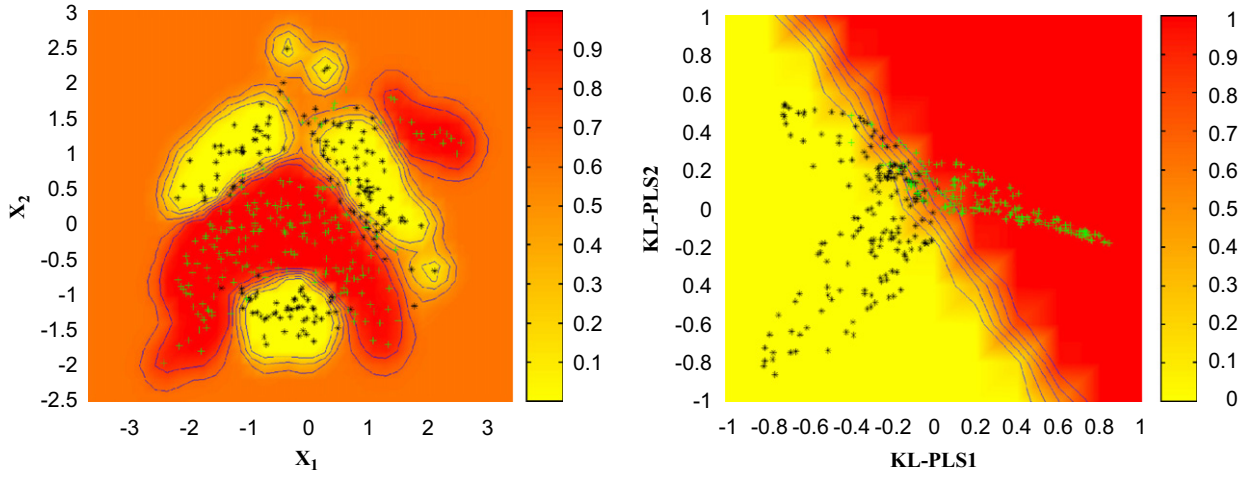

Fig. 2. Banana depicted onto the original space and the two first KL-PLS components. Colors symbolize conditional probability $P(y = 1/X = x)$.

classification accuracy is then observed as a function of the percentage of noise injected in the target. Fig. 4 shows that KL-PLS performs as well as SVM$^{\text{light}}$ (Joachims, 1999) in this example.

The test set classification accuracy curves for SVM and KL-PLS have a similar shape. It is remarkable that the accuracy is almost unchanged up to 30% degradation.

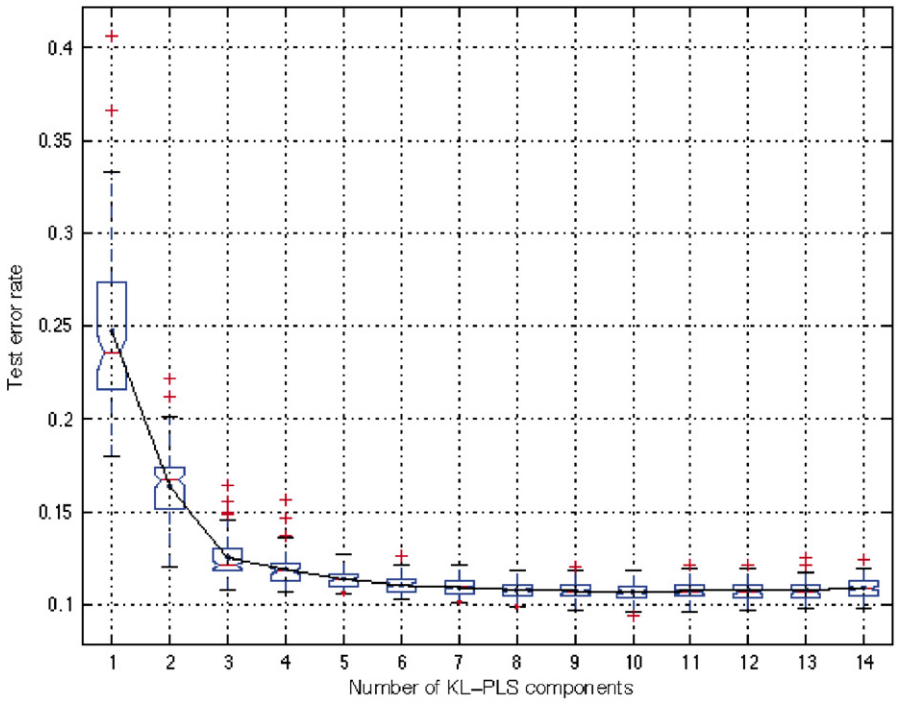

Fig. 3. Test error as a function of the number of KL-PLS components retained for Banana data. Boxplots show the distribution of the 100 individual test error rates. The black points represent the means and the horizontal red lines the median.

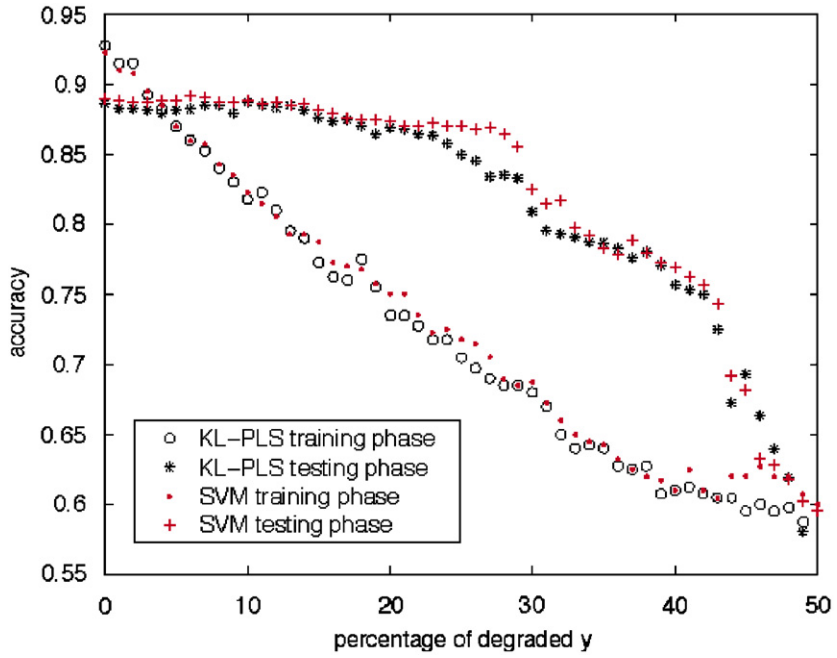

Fig. 4. Comparison between KL-PLS ($m = 10$ and $sigma = 0.9$) and SVM ($C = 1$ and $sigma = 0.9$) test set accuracy rate as a function of percentage of degraded y.

6.3. KL-PLS and high-dimensional data

The ovarian cancer and the lung cancer datasets from Li and Liu (2002) were chosen to test the efficiency of KL-PLS for high-dimensional data. Gene expression profiles obtained from DNA micro-arrays are used to classify different types of tumors. In these situations, there is generally a huge number of variables and a small number of observations: ovarian (respectively, lung) is a 253×15154, (respectively, $181 \times 12\,533$) data set with two classes. The evaluation

Table 3
Comparison of classification errors (mean $\pm$ standard deviation) (test set) for KL-PLS and SVM (Shen and Tan, 2005)

Data set	KL-PLS	SVM
Ovarian	$0.0 \pm 0.0^{*}$	0.22 ± 0.5
Lung	$0.24 \pm 0.49^{*}$	0.83 ± 0.82

$^{*}p < 0.05$.

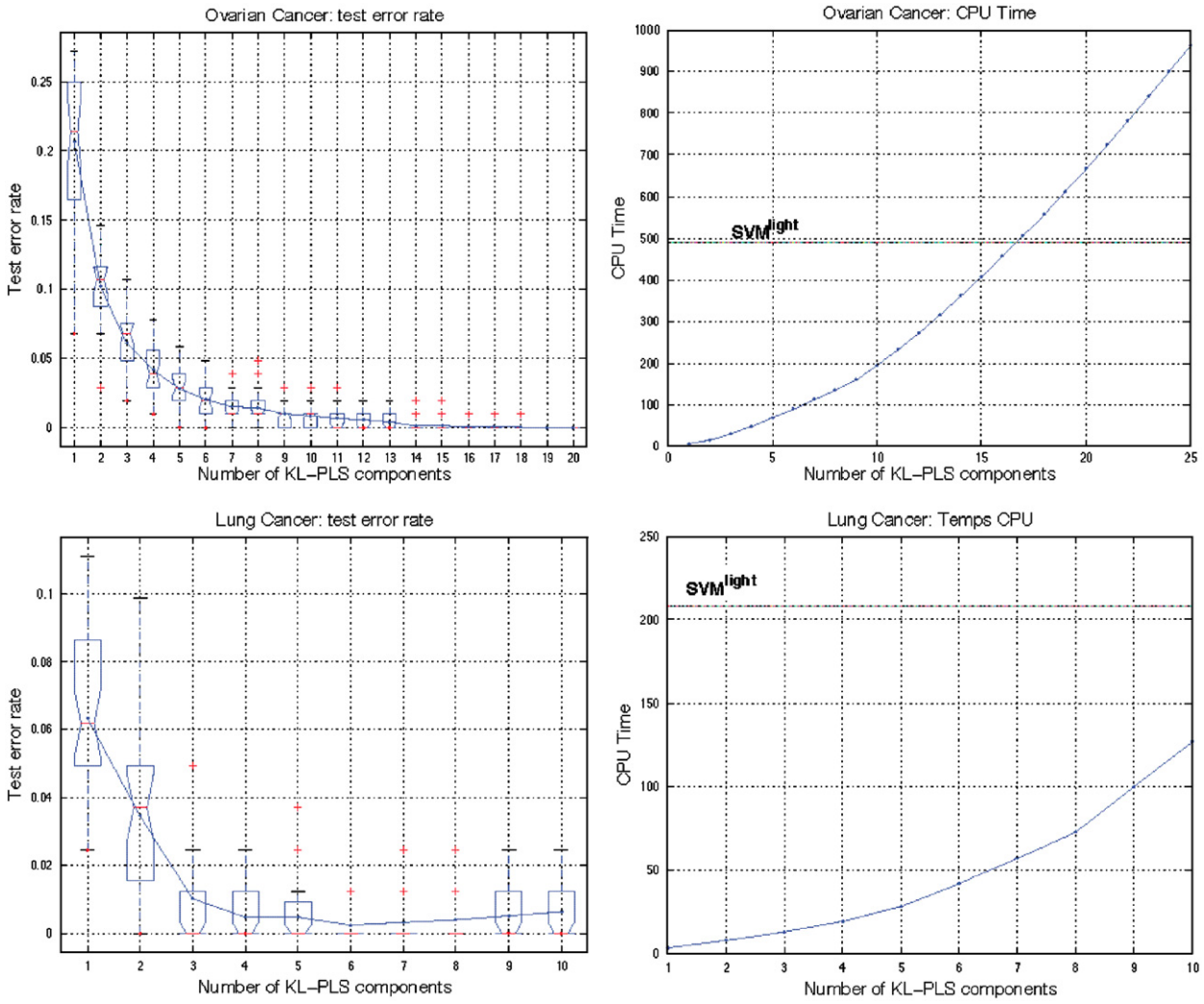

Fig. 5. Test error rate (respectively, CPU time) of KL-PLS as the function of the number of retained components for ovarian data and lung data (left) (respectively, (right)). Horizontal lines show the CPU time of $\text{SMV}^{\text{light}}$. Same legend for the boxplots as in Fig. 3.

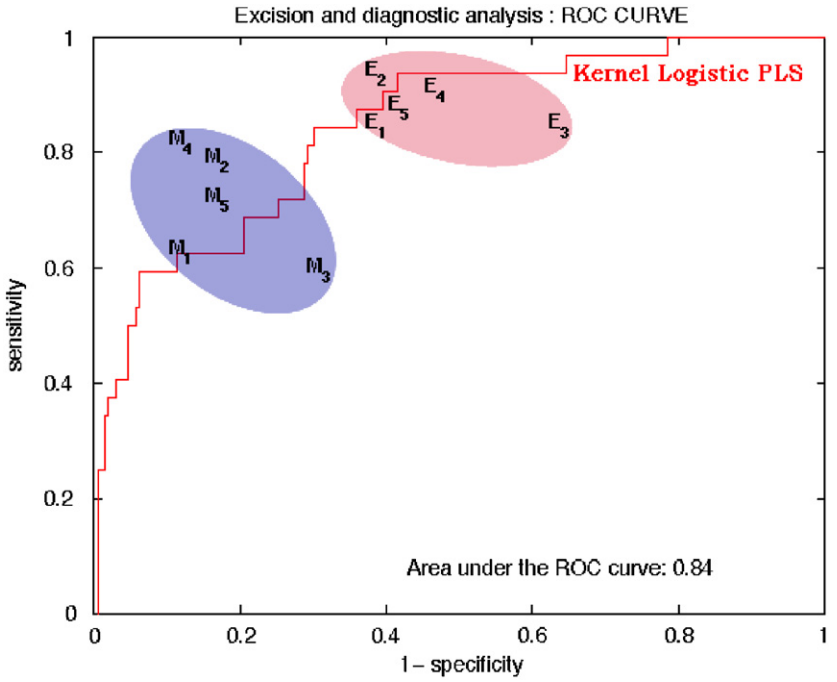

Fig. 6. Roc curve of KL-PLS. M_i: Accuracy of dermatologist i in terms of sensitivity and (1-specificity) for the melanoma diagnosis. E_i: Accuracy of dermatologist i in terms of sensitivity and (1-specificity) for the excision decision.

of KL-PLS on these data sets is designed as follows: sizes of training samples and testing samples are set to 150 and 103 (ovarian) and 100 and 81 (lung). Thirty random partitions are then carried out. Average testing errors with their standard deviations are then computed.

As it frequently happens for high-dimensional problems (e.g. text mining, genomic), a linear kernel is effective in performing an efficient discrimination. Therefore, the number of KL-PLS components is the unique parameter to adjust. Here, its value is based on the minimum classification error observed by cross validation on the first five training sets. It is set to 19 for ovarian and six for lung. Linear SVM classifier results were obtained by Shen and Tan, 2005. Results of KL-PLS and SVM are reported in Table 3.

To test whether KL-PLS is statistically more accurate than SVM on ovarian and lung dataset, T-test between the means of the testing errors of the two classifiers is used. From these experiments, it appears that KL-PLS outperforms SVM. Nevertheless, we can note that the error rates observed are low for both methods and are always due to a few observations only.

We may emphasize in these examples that the $n \ll p$ configuration is particularly convenient for KL-PLS. The construction of the KL-PLS components is computationally efficient because it is obtained from a "small" $n \times n$ kernel matrix.

Fig. 5 shows the test error rate and the CPU time of KL-PLS as a function of the number of retained KL-PLS components for ovarian cancer and lung cancer.

Horizontal lines provide the CPU time of SVM$^{\text{light}}$. For lung cancer, the CPU time of KL-PLS with six components (Matlab 6.5 implementation) is about five times lower than the CPU time of SVM$^{\text{light}}$ (C implementation). The CPU time of KL-PLS with 19 components for ovarian cancer is greater than the CPU time of SVM$^{\text{light}}$. Nevertheless, at equal error rate (obtained with 14 components), the CPU time of KL-PLS is 30% lower than the CPU time of SVM$^{\text{light}}$.

6.4. Melanoma detection from digital medical imaging

In this section, we applied KL-PLS to a medical problem based on skin tumor detection (benign lesion vs. melanoma). From 227 images of pigmented skin lesion provided by dermatology departments of British Hertfort Hospital and

Louis Mourier Hospital, we have extracted five geometric and 38 colorimetric features. Each lesion is thus characterized by 43 variables. Thirty-two melanoma and 195 benign lesions make up the database. The objective of this application is to provide a diagnosis based on KL-PLS (melanoma detection) that can be compared with the diagnoses of five senior dermatologists. The histology (melanoma lesion or benign lesion) is the variable to predict (gold standard).

In medical diagnosis applications, accuracy is not the main issue. Sensitivity and specificity (or predictive positive and negative values) are far more relevant because of different misclassification costs. Fig. 6 provides, in addition to dermatologist's performances, the ROC curve of KL-PLS. See Hanley (1989) for details on the ROC curve.

Each dermatologist is characterized by two points: the first point (respectively, the second point) M_i (respectively, E_i) corresponds to the accuracy of diagnosis (respectively, accuracy of excision decision) for dermatologist i (in terms of sensitivity and 1-specificity). We may observe that KL-PLS reaches performances close to the dermatologist's performances in terms of diagnosis and excision decision. Results are obtained by leave-one-out cross validation technique. We have retained three components with $\sigma = 100$. KL-PLS provides, for each tested lesion, a probability to be a melanoma and the ROC curve is built from these probabilities.

We highlight that the output probability is useful here because it gives rise to the possibility to reproduce the dermatologist's behavior. Indeed, diagnostic and therapeutic decisions correspond to a modulation of the sensitivity level. This decision flexibility is very important for the melanoma detection problem because of the unbalanced misclassification cost: sensitivity is far more important than specificity since disregarding a melanoma is a severe error.

7. Conclusion

This work demonstrates competitiveness of KL-PLS with other state-of-the art classification methods for 11 benchmark discrimination data sets as well as for three medical classification problems. The algorithm is simple to implement since it is mainly composed of ordinary least squares and logistic regressions. The complexity control achieved by the supervised dimensionality reduction performed by KL-PLS offers a potentially fruitful alternative to Tikhonov regularization based methods. Indeed, the projection of the high-dimensional data onto a small number of KL-PLS components allows visual inspection of data layout. It provides clues about data structure and selection of kernel parameter(s). Visualization of the spatial relationships between classes helps evaluating quality of classification and feature space; Are classes linearly separables? How well-separated and coherent are they? Finally, KL-PLS offers an estimate of the conditional probability, often of interest itself, in medical field for example.

We would like to point out that the direct factorization of the kernel matrix in the framework of KL-PLS presents two significant advantages:

(i) Rectangular kernels can be factorized by KL-PLS. Derived from the fact that KL-PLS has only to handle a $n \times n$ matrix, the algorithmic limitation is much more connected to the number of observations than to the number of variables. Therefore, this approach allows management of very high-dimensional data. Large training set size can also be managed, thanks to the sampling that can be performed within KL-PLS with no modification of the algorithm.

(ii) The requirement of semi-definiteness may rule out the most natural pairwise similarity functions for a given problem. In the KL-PLS setting, only similarity measures are required and there is no need defining a dot product in the feature space; the kernel matrix need not verify the Mercer's conditions (positive definite).

Extension of KL-PLS to the multi-class problem is easy for ordinal output y (see Appendix B) and more complicated for categorical output (work in progress). Validation of the Kernel Generalized PLS, a nonlinear version of the PLS-GLR that could be derived from a generalization of DKPLS, is an interesting perspective.

Acknowledgments

The Ph.D. fellowship of Arthur Tenenhaus is granted by KXEN (Knowledge eXtraction ENgines), a global analytic software company and the National Association of Technical Research. We thank Fanny Aziza, Michel Tenenhaus, Léon Bottou and Yann Lecun for helpful and fruitful discussions.

Appendix A. PLS generalized linear regression algorithm

Algorithm 1. PLS generalized regression (PLS-GLR)

Step 1. ***Computation of the PLS-GLR components***

Computation of the first PLS-GLR component t_1

1. Computation of the coefficients a_{1j} for each simple generalized linear regression of y on x_j, $j = 1, \ldots, p$.
2. Normalization of the column vector a_1 made by a_{1j}'s:$w_1 = a_1/\|a_1\|$.
3. Computation of the first PLS-GLR component as $t_1 = Xw_1$.

Computation of the hth *PLS-GLR component* t_h

1. Computation of the residual $x_{h-1,1}, \ldots, x_{h-1,p}$ from the multiple regression of x_j, $j = 1, \ldots, p$ on $t_1, \ldots, t_{h-1}$. Let $X_{h-1} = [x_{h-1,1}, \ldots, x_{h-1,p}]$.
2. Computation of the coefficients a_{hj} of $x_{h-1,j}$ in the generalized linear regression of y on $t_1, \ldots, t_{h-1}$ and each $x_{h-1,j}$, $j = 1, \ldots, p$.
3. Normalization of the column vector a_h made by a_{hj}'s: $w_h = a_h/\|a_h\|$.
4. Computation of the hth PLS-GLR component as $t_h = X_{h-1}w_h$.
5. Expression of the component t_h in terms of X as $t_h = Xw_h^*$.

Step 2. ***Generalized linear regression of*** y ***on the*** m ***retained PLS-GLR components***

Appendix B. Kernel logistic PLS algorithm

Algorithm 2. Kernel Logistic PLS (KL-PLS)

Step 1. **Computation of the kernel matrix:** K

Step 2. **Computation of the KL-PLS components**

Computation of the first KL-PLS component t_1

1. Logistic regression of y on each k_j, $j = 1, \ldots, n \Rightarrow a_{1j}$.
2. Normalization of a_1 made by a_{1j}'s: $w_1 = a_1/\|a_1\|$.
3. The first KL-PLS component is $t_1 = Kw_1$.

Computation of the hth *KL-PLS component* t_h.

1. Regression of each k_j, $j = 1, \ldots, n$ on $t_1, \ldots, t_{h-1}$ $\Rightarrow K_{h-1} = [k_{h-1,1}, \ldots, k_{h-1,n}]$
2. Logistic regression of y on $t_1, \ldots, t_{h-1}$ and each $k_{h-1,j}$, $j = 1, \ldots, n \quad \Rightarrow a_{hj}$.
3. Normalization of a_h made by a_{hj}'s: $w_h = a_h/\|a_h\|$.
4. The hth KL-PLS component is $t_h = K_{h-1}w_h$.
5. Expression of the component t_h in terms of K as $t_h = Kw_h^*$.

Step 3. **Logistic regression of** y**on the** m **retained KL-PLS components**

By replacing binary logistic regressions by ordinal logistic regressions using proportional odds model, we may note that the KL-PLS algorithm is still convenient when the response y is ordinal.

References

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

Allison, D.P., 1999. Logistic regression using the SAS system: theory and application, SAS institute, ISBN: 1-58025-352-0.

Bach, F.R., Jordan, M.I., 2005. Predictive low-rank decomposition for kernel methods. In: Proceedings of the Twenty-second International Conference on Machine Learning, Bonn, Germany.

Baffi, G., Martin, E.B., Morris, A.J., 1999. Non-linear projection to latent structures revisited (the neural networks PLS algorithm). Comput. Chem. Eng. 23, 1293–1307.

Barker, M., Rayens, W.S., 2003. Partial least squares for discrimination. J. Chemometrics 17, 166–173.

Bastien, P., Vinzi, V.E., Tenenhaus, M., 2005. PLS generalized linear regression. Comput. Statist. Data Anal. 48, 17–46.

Bennett, K.P., Embrechts, M.J., 2003. An optimization perspective on kernel partial least squares regression. Advances in Learning Theory: Methods, Models and Applications. NATO Sciences Series III: Computer and Systems Sciences, vol. 190. IOS Press, Amsterdam, pp. 227–250.

Boser, B.E., Guyon, I., Vapnik, V.N., 1992. A training algorithm for optimal margin classifiers. In: Proceedings of the Fifth Annual Workshop on Computational Learning Theory, vol. 5, Pittsburgh, pp. 144–152.

Burges, C.J.C., 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2 (2), 121–167.

Cherkassky, V., Mulier, F.P., Vapnik, V.N., 1999. Model complexity control for regression using VC generalization bounds. IEEE Transactions on Neural Networks 10, 1075–1089.
Cortes, C., Vapnik, V., 1995. Support vector network. Mach. Learning 20, 273–297.
Evgeniou, T., Pontil, M., Poggio, T., 1999. Regularization networks and support vector machines. Adv. Comput. Math. 13, 1–50.
Firth, D., 1993. Bias reduction of maximum likelihood estimates. Biometrika 80, 27–38.
Garthwaite, P.H., 1994. An interpretation of partial least squares. J. Amer. Statist. Assoc. 89 (425), 122–127.
Hanley, J.A., 1989. Receiver operating characteristic methodology: the state of the art. Critical Reviews in Diagnostic Imaging 29, 307–335.
Heinze, G., Schemper, M., 2002. A solution to the problem of separation in logistic regression. Statist. Medicine 21, 2409–2419.
Höskuldsson, A., 1988. PLS regression methods. J. Chemometrics 2, 211–228.
Joachims, J., 1999. Making large-scale SVM learning practical. In: Schölkopf, B., Burges, C., Smola, A. (Eds.), Advances in Kernel Methods—Support Vector Learning. MIT Press, Cambridge, MA.
Li, J., Liu, H., 2002. Kent Ridge Biomedical Dataset Repository. ⟨http://sdmc-lit.org.sg/GEDatasets⟩
Rätsch, G., Onoda, T., Muller, K.R., 2001. Soft margin for adaboost. Mach. Learning 42, 287–320.
Rosipal, R., Trejo, L.J., 2001. Kernel partial least squares regression in reproducing kernel hilbert space. J. Mach. Learning Res. 2, 97–123.
Rosipal, R., Trejo, L.J., Matthews, B., 2003. Kernel PLS-SVC for linear and nonlinear classification. In: Proceeding of the Twentieth International Conference on Machine Learning, Washington, USA.
Schölkopf, B., Smola, A.J., 2002. Learning With Kernel—Support Vector Machines Regularization Optimization and Beyond. MIT Press, Cambridge, MA.
Schölkopf, B., Smola, A.J., Müller, K.R., 1998. Nonlinear component analysis as a kernel eigenvalue problem. Neural Comput. 10, 1299–1319.
Shawe-Taylor, J., Cristianini, N., 2004. Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge.
Shen, L., Tan, E.C., 2005. Dimension reduction-based penalized logistic regression for cancer classification using microarray data. IEEE/ACM Trans. Comput. Biol. Bioinformatics 2 (2), 166–175.
Smola, A.J., Schölkopf, B., 2000. Sparse greedy matrix approximation for machine learning. In: Proceeding of the Seventeenth International Conference on Machine Learning, Stanford, USA.
Tenenhaus, A., 2002. La Régression Logistique PLS validée par bootstrap. Master degree of statistics, Pierre and Marie Curie University, Paris, France.
Tenenhaus, M. , 1998. La Régression PLS, éditions Technip.
Tenenhaus, M., 2005. La régression logistique PLS. In: Droesbeke, J.J., Lejeune, M., Saporta, G. (Eds.), Modèles statistiques pour données qualitatives. Editions Technip.
Tenenhaus, A., Giron, A., Saporta, G., Fertil, B., 2005. Kernel logistic PLS: a new tool for complex classification. In: 11th International Symposium on Applied Stochastic Models and Data Analysis, Brest, France.
Tikhonov, A.N., Arsenin, V.Y., 1977. Solution of Ill-Posed Problem. Wiley, New York.
Tucker, L.R., 1958. An inter-battery method of factor analysis. Psychometrika 23 (2),
Vapnik, V., 1998. Statistical Learning Theory. Wiley, New York.
Wahba, G., 1999. Support vector machines, reproducing kernel Hilbert spaces and randomized GACV. In: Advances in Kernel Methods—Support Vector Learning, pp. 69–88.
Webb, A.R., 1996. Nonlinear feature extraction with radial basis functions using a weighted multidimensional scaling stress measure. In: 13th International Conference on Pattern Recognition.
Williams, C.K.I., Seeger, M., 2000. Effect on the input density distribution on kernel-based classifiers. In: Proceeding of the Seventeenth International Conference on Machine Learning, Stanford, USA.
Wold, S., 1992. Non-linear partial least squares modelling. II Spline inner function. Chemometrics Intell. Lab. Systems 14, 71–84.
Wold, S., Martens, H., Wold, H., 1983. The multivariate calibration problem in chemistry solved by the PLS method. In: Lecture Notes in Mathematics.Springer, Heidelberg, pp. 286–293.
Wold, S., Kettaneh-Wold, N., Skagerberg, B., 1989. Non-linear PLS modelling. Chemometrics Intell. Lab. Systems 7, 53–65.
Wu, W., Massart, D.L., de Jong, S., 1997. The kernel PCA algorithms for wide data. Part II: fast cross validation and application in classification of NIR data. Chemometrics Intell. Lab. Systems 37, 271–280.
Zhu, J., Hastie, T., 2005. Kernel logistic regression and the import vector machine. J. Comput. Graphical Statist. 14 (1), 185–205.